

THE NATA CODE - PROGRAMMER'S MANUAL

NASA CR-

141744

JSC-III03

W. L. Bade and J. M. Yos
AVCO SYSTEMS DIVISION
201 Lowell Street
Wilmington, Massachusetts 01887

NAS9-9744
CR-141744
2 September 1975
Final Report
Volume III

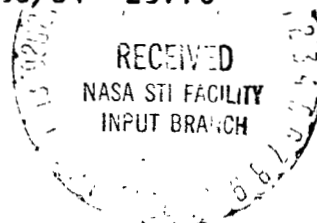
(NASA-CF-141744) THE NATA CODE; THEORY AND
ANALYSIS. VOLUME 3: PROGRAMMER'S MANUAL
Final Report (Avco Systems Div., Wilmington,
Mass.) 635 p HC \$16.25 CSCI 20D

876-21458

63/34 Unclass
25776

Prepared for

LYNDON B. JOHNSON SPACE CENTER
Houston, Texas 77058



1. Report No. NASA CR-141744		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle THE NATA CODE - PROGRAMMER'S MANUAL, Volume III				5. Report Date 2 September 1975	
				6. Performing Organization Code JSC-11103	
7. Author(s) W. L. Bade and J. M. Yos				8. Performing Organization Report No. AVSD-0138-75-RR	
				10. Work Unit No.	
9. Performing Organization Name and Address Avco Systems Division 201 Lowell Street Wilmington, Massachusetts 01887				11. Contract or Grant No. NAS9-9744	
				13. Type of Report and Period Covered Contractor Report Final Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058				14. Sponsoring Agency Code	
15. Supplementary Notes The final report on Contract NAS9-9744 consists of three volumes: Vol. I The NATA Code - Theory and Analysis (NASA CR-2547) Vol. II The NATA Code - User's Manual (NASA CR-141743) Vol. III The NATA Code - Programmer's Manual (NASA CR-141744) (present document)					
16. Abstract <p>The NATA code is a computer program for calculating quasi-one-dimensional gas flow in axisymmetric and two dimensional nozzles and rectangular channels. The code also computes stagnation point conditions on axisymmetric or two-dimensional models and the conditions on blunt wedges inserted into the flow. The theory and analysis underlying the calculations have been documented in Volume I of this final report. Code inputs, outputs, and precoded data have been defined in Volume II, which is a user's manual. The present, third volume of the final report is a programmer's manual for the code. It provides a listing of the Fortran IV source program; a complete glossary of Fortran symbols; a discussion of the purpose and method of operation of each subroutine (including mathematical analyses of special algorithms); and a discussion of the operation of the code on IBM/360 and UNIVAC 1108 systems, including required control cards and the overlay structure used to accommodate the code to the limited core size of the 1108. In addition, this volume provides similar information to document the programming of the NOZFIT code, which is employed to set up nozzle profile curvefits for use in NATA.</p> <p style="text-align: center;">ORIGINAL PAGE IS OF POOR QUALITY</p>					
17. Key Words (Suggested by Author(s)) Nonequilibrium flow Nozzle flow Arc-heated wind tunnels Stagnation-point heat transfer Wedge heat transfer Laminar boundary layer Thermal nonequilibrium			18. Distribution Statement STAR Subject Category: 34 (Fluid Mechanics and Heat Transfer)		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 637	
				22. Price* \$16.25	

PREFACE

This is the third volume of a final report documenting the development of a computer program (NATA) for calculating the flow in arc-heated wind tunnels and the conditions on models tested in such reentry simulation facilities. Volume I of the report provides a general summary of the code's capabilities and presents the theory and analysis underlying its operation. Volume II is a user's manual which defines NATA's inputs, outputs, and precoded data on standard gas models and nozzle geometries. The present volume is a programmer's manual for NATA. It documents the programming of NATA in sufficient detail to support code modifications, and provides information needed for running the code on the UNIVAC 1108 under the EXEC II system, or on the IBM 360/75 under HASP II.

TABLE OF CONTENTS

	<u>Page</u>
1. Introduction	1
2. Structure and Operation	2
2.1 Brief Description of the NATA Code	2
2.2 Subroutines and Linkage	2
2.3 UNIVAC 1108 Version	12
2.4 IBM 360 Version	16
3. Analysis of Routines	20
3.1 Main Program	20
3.2 Subroutine AESØLN	28
3.3 Subroutine AGSØLN	30
3.4 Subroutine AXFIT	32
3.5 Subroutine AXSECT	33
3.6 Subroutine BLAYER	33
3.7 Subroutine BLCALL	44
3.8 Subroutine BXSECT	45
3.9 Subroutine CØMM	46
3.10 Subroutine CXSECT	61
3.11 Subroutine DERIVS	61
3.12 Subroutine DSMSØL	62
3.13 Subroutine DUMP	64
3.14 Subroutine DUMPEX	64
3.15 Subroutine ELCØND	65
3.16 Subroutine ELTIME	65
3.17 Subroutine EPART	66
3.18 Subroutine EQCALC	67
3.19 Subroutine EXACT	70
3.20 Function EXP	73
3.21 Subroutine FINDX	76
3.22 Subroutine FRØZEQ	77
3.23 Subroutine GEØM	91
3.24 Subroutine GEØMAR	101
3.25 Subroutine INGAS	103
3.26 Subroutine INIT	105
3.27 Subroutine KANDMU	105

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
3.28 Subroutine KINT	106
3.29 Subroutine LIST	107
3.30 Subroutine MATINV	108
3.31 Subroutine MØDEL	108
3.32 Subroutine NEWRAP	115
3.33 Subroutine NEXTMP	120
3.34 Subroutine NØNEQ	121
3.35 Subroutine NRMAX	139
3.36 Subroutine ØUT	139
3.37 Subroutine ØUT1	140
3.38 Subroutine PERT	148
3.39 Function PIØMEG	149
3.40 Subroutine PRØP.....	150
3.41 Subroutine PRTA	150
3.42 Subroutine PUTQIN	151
3.43 Subroutine QCØUL	154
3.44 Subroutine QEX	155
3.45 Subroutine QEXP	156
3.46 Subroutine QINTRP	156
3.47 Subroutine Q1J	157
3.48 Subroutine QMIX	157
3.49 Subroutine QREPP	158
3.50 Subroutine QSAME	159
3.51 Subroutine QTAB	159
3.52 Subroutine Q11	159
3.53 Subroutine Q12	159
3.54 Subroutine Q13	160
3.55 Subroutine Q14	160
3.56 Subroutine RADIUS	160
3.57 Subroutine READ	161
3.58 Subroutine RESET	166
3.59 Subroutine RESTMP	166
3.60 Subroutine RNKT	174
3.61 Subroutine SHØCK	178
3.62 Subroutine SIMQ	181
3.63 Subroutine STUN1S	181
3.64 Subroutine THERM	186

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
3.65 Subroutine THRØAT	188
3.66 Subroutine TRANSP	188
3.67 Subroutine WEDGE	192
3.68 Subroutine WESØLN	193
3.69 Subroutine XSECT	195
 4. Glossary of Fortran Symbols	 212
4.1 Unlabelled Common	212
4.2 Common /AEGEØM/	224
4.3 Common /AREA/	224
4.4 Common /AVG/	225
4.5 Common /BL/	225
4.6 Common /BLNE/	226
4.7 Common /BLØUT/	227
4.8 Common /BLRAD/	227
4.9 Common /CERRØR/	227
4.10 Common /CHAN/	227
4.11 Common /CØLDSP/	228
4.12 Common /CØNVRT/	228
4.13 Common /DRV/	229
4.14 Common /EELEM/	229
4.15 Common /ELEM/	230
4.16 Common /EQC/	230
4.17 Common /EQC2/	230
4.18 Common /ERRØR/	231
4.19 Common /GLIM2/	231
4.20 Common /INGNE/	231
4.21 Common /LN/	231
4.22 Common /MASSFL/	231
4.23 Common /MIXT/	232
4.24 Common /MØDPAR/	233
4.25 Common /MØDPT/	234
4.26 Common /MØDP2/	234
4.27 Common /NEQ/	235
4.28 Common /NEWMP/	235
4.29 Common /NØZZ/	235
4.30 Common /ØUTMØD/	236

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
4.31 Common /ØUTPUT/	237
4.32 Common /PØLYAT/	239
4.33 Common /RDLIST/	239
4.34 Common /RDMAIN/	239
4.35 Common /RDMØD/	240
4.36 Common /RDØUT/	240
4.37 Common /RDTR/	240
4.38 Common /RDWEDG/	240
4.39 Common /REAC/	241
4.40 Common /READAT/	242
4.41 Common /RENE/	245
4.42 Common /RESRP/	245
4.43 Common /SPEC/	246
4.44 Common /SS/	247
4.45 Common /STAG/	247
4.46 Common /STEPS/	247
4.47 Common /SWITCH/	247
4.48 Common /TAPØUT/	248
4.49 Common /TEMPRY/	248
4.50 Common /THRT/	248
4.51 Common /TNCE/	249
4.52 Common /TNDC/	249
4.53 Common /TNE/	249
4.54 Common /TNEQ/	250
4.55 Common /TNERK/	251
4.56 Common /TNØNEQ/	253
4.57 Common /TRANI/	254
4.58 Common /TRANJ/	254
4.59 Common /TRANS1/	254
4.60 Common /TRANS2/	255
4.61 Common /TRANS3/	256
4.62 Common /TRANS4/	256
4.63 Common /TRANS5/	256
4.64 Common /TRANS7/	257
4.65 Common /TRANS8/	257
4.66 Common /TRANV/	257
4.67 Common /TRPRØP/	258
4.68 Common /WMØD/	258

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
4.69 Main Program	258
4.70 Subroutine AESØLN	260
4.71 Subroutine AGSØLN	261
4.72 Subroutine AXFIT	262
4.73 Subroutine AXSECT	262
4.74 Subroutine BLAYER	263
4.75 Subroutine BLCALL	270
4.76 Subroutine BXSECT	270
4.77 Subroutine CØMM	270
4.78 Subroutine CXSECT	274
4.79 Subroutine DERIVS	2
4.80 Subroutine DSMSØL	27
4.81 Subroutine DUMP	276
4.82 Subroutine DUMPEX	276
4.83 Subroutine ELCØND	276
4.84 Subroutine ELTIME	276
4.85 Subroutine EPART	277
4.86 Subroutine EQCALC	277
4.87 Subroutine EXACT	278
4.88 Function EXP	279
4.89 Subroutine FINDX	279
4.90 Subroutine FRØZEQ	280
4.91 Subroutine GEØM	284
4.92 Subroutine GEØMAR	285
4.93 Subroutine INGAS	286
4.94 Subroutine INIT	287
4.95 Subroutine KANDMU	287
4.96 Subroutine KINT	288
4.97 Subroutine LIST	289
4.98 Subroutine MATINV	290
4.99 Subroutine MØDEL	291
4.100 Subroutine NEWRAP	295
4.101 Subroutine NEXTMP	296
4.102 Subroutine NØNEQ	296
4.103 Subroutine NRMAX	301
4.104 Subroutine ØUT1	301

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
4.105 Subroutine PERT	304
4.106 Function PIØMEG	305
4.107 Subroutine PRØP	305
4.108 Subroutine PRTA	305
4.109 Subroutine PUTQIN	305
4.110 Subroutine QCØUL	306
4.111 Subroutine QEX	306
4.112 Subroutine QEXP	307
4.113 Subroutine QINTRP	308
4.114 Subroutine QLJ	308
4.115 Subroutine QMIX	309
4.116 Subroutine QREPP	309
4.117 Subroutine QSAME	309
4.118 Subroutine QTAB	310
4.119 Subroutine Q11	310
4.120 Subroutine Q12	311
4.121 Subroutine Q13	311
4.122 Subroutine Q14	312
4.123 Subroutine RADIUS	312
4.124 Subroutine READ	313
4.125 Subroutine RESET	317
4.126 Subroutine RESTMP	317
4.127 Subroutine RNKT	319
4.128 Subroutine SHØCK	322
4.129 Subroutine SIMQ	324
4.130 Subroutine STUNTS	325
4.131 Subroutine THERM	327
4.132 Subroutine THRØAT	329
4.133 Subroutine TRANSP	329
4.134 Subroutine WEDGE	330
4.135 Subroutine WESØLN	337
4.136 Subroutine XSECT	338
 5. References to Common Variables	 345
6. Listing of the NATA Code	374

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS
(Continued)

	<u>Page</u>
Appendix - Programmer's Manual for the NOZFIT Code	576
A.1 Analysis of NOZFIT Routines	576
A.2 Glossary of Fortran Symbols for NOZFIT	594
A.3 Listing of NOZFIT	608
References	624

LIST OF ILLUSTRATIONS

	<u>Page</u>
1. Overlay Structure for UNIVAC 1108 Version of NATA ..	14
2. Control Cards for Generating a NATA PCF Tape on the UNIVAC 1108	15
3. Control Cards for Running NATA on the UNIVAC 1108 ..	15
4. Control Cards for Running NATA on the IBM 360	19
5. Flowchart of NATA Main Program	21
6. Flowchart of Subroutine BLAYER	35
7. Flowchart of Main Loop Over Reactions in Subroutine COMM	50
8. Flowchart of the Solution for the Derivatives of the Flow Variables in the Numerical Integration	75
9. Flowchart for the Determination of Sonic Conditions.	79
10. Three Cases in the Algorithm for Finding the Throat Temperature	81
11. Flowchart of Subroutine FRØZEQ	83
12. Flowchart of Subroutine GEØM	93
13. Approximations to the Density-Area Ratio Relation ..	95
14. Overall Flowchart of Subroutine MØDEL	110
15. Flowchart of the Normal Shock Solution	113
16. Flowchart of the Solution for Conditions at the In- viscid Stagnation Point	116
17. Flowchart of Subroutine NØNEQ	122

LIST OF ILLUSTRATIONS
(Continued)

	<u>Page</u>
18. Flowchart of Initialization Section of Subroutine NØNEQ	124
19. Flowchart of Switch to Downstream Region in Sub- routine NØNEQ	126
20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine NØNEQ	127
21. Flowchart of Perturbation Solution and Switch to In- tegration in Subroutine NØNEQ	129
22. Flowchart of Controls for the Nonequilibrium Inte- gration in Subroutine NØNEQ	134
23. Overall Flowchart of Subroutine ØUT1	141
24. Flowchart of the Operations Produced by Calling Entry ØUT2 of Subroutine ØUT1	143
25. Iteration Scheme of Third Option for Specifying Reservoir Conditions	172
26. Flowchart of Subroutine RNKT	175
27. Schematic Flowchart of Subroutine XSECT	197
28. Geometry of the Condition ICØND = 1	578
29. Geometry of the Condition ICØND = 2	580
30. Geometry of the Condition ICØND = 3	582
31. Geometry of the Condition ICØND = 4	584

LIST OF TABLES

	<u>Page</u>
I. Derivatives Determined by EXACT During the Numerical Integration	74
II. Subroutines Used in Cross Section Computations	153
III. Indices Used for Calculation of Effective Coulomb Cross Sections in Subroutine XSECT	209

THE NATA CODE - PROGRAMMER'S MANUAL

By W. L. Bade and J. M. Yos
Avco Systems Division
Wilmington, Massachusetts

1. INTRODUCTION

The NATA code is a computer program for solving problems of steady, quasi-one-dimensional gas flow in nozzles. The code's capabilities, and the theory and analysis underlying its operation, have been documented in the first volume of this report (ref. 1). The inputs, outputs, and precoded data on standard gas models and nozzle geometries have been defined in the second volume (ref. 2). The present, third volume provides information required for running NATA on UNIVAC 1108 and IBM 360 computers, and documents the programming of NATA so that programmers and computer-oriented users will be able to evaluate the techniques used, maintain the code in the face of operating-system changes, convert it for use on other types of computer, introduce modifications and correct errors.

Section 2 provides a general description of NATA, including a catalog of its subroutines with a statement of function and a list of calling routines for each. This section also gives sample control-card decks for running the code on the UNIVAC 1108 under the EXEC II system, and on the IBM 360/75 under HASP II and CS 21.

Section 3 discusses each subroutine, explaining its purpose in the code and its method of operation. Many special algorithms are analyzed. Where appropriate, flowcharts are used to display the logic and clarify the methods used. The Fortran variables stored in common blocks and used in the individual routines are all defined in a glossary of symbols (Section 4). Section 5 lists the subroutines in which each common variable is set, used, or referenced as a subroutine argument. Finally, a complete listing of the source program is given in Section 6.

An appendix provides similar documentation of the NOZFIT code, a much smaller program provided for setting up nozzle profile curvefits for use in NATA. A user's manual for NOZFIT is included in Volume II (ref. 2). The appendix includes documentation of a general-purpose plotting subroutine (GRAPH) for producing plots on S-C 4020, S-C 4060, or other plotting equipment using the NAA software package for the SC-4020 (ref. 3).

2. STRUCTURE AND OPERATION

2.1 Brief Description of the NATA Code

NATA is a Fortran IV program consisting of a main program, two block data routines, and 66 subroutines. The source deck contains approximately 8500 cards. The program exists in two versions, one for use on the IBM 360 system, the other for the UNIVAC 1108. The IBM version is entirely in double precision, whereas the UNIVAC version is a single precision program with some double precision arrays and subroutines. One version can be converted into the other by inserting or removing most of the IMPLICIT REAL*8 (A-H, \emptyset -Z) cards which type all floating-point variables as double precision in the IBM 360 version.*

The IBM 360 version requires about 410K bytes of core storage, including buffers. The UNIVAC version is run on the 1108 using overlay, and requires about 123000₈ or 42500₁₀ words of storage. It fits into the two-bank processors at NASA/JSC with about 2000 words to spare.

2.2 Subroutines and Linkage

A list of the routines of NATA is presented below. Names of entry points are included, and are enclosed in parentheses to distinguish them from subroutine and function names. The function performed by each routine or entry point is described briefly, and in each case a complete list of calling routines is included. The order of the list is as follows: main program; block data routines; subroutines, functions, and entry points in alphanumeric order.

*Also, the IBM version contains two extra subroutines, EXP and RESET, for reasons explained below.

MAIN	Main program; calls other routines to perform the required calculations.
BLKD1	BLØCK DATA routine containing precoded data on elements, species, reactions, gas models, and nozzle geometri s.
BLKD2	BLØCK DATA routine containing precoded data on transport-property cross sections.
AESØLN	Calculates the effective area ratio A_e and its derivative dA_e/dx from data on nozzle geometry and the boundary layer displacement thickness.* Called by GEØM.
AGSØLN	Calculates the geometric area ratio A_g and the axial coordinate x from data on the effective area ratio A_e and the boundary layer displacement thickness. Called by FRØZEN, EQUIL, and AXFIT.
AXFIT	Calculates the axial coordinate x corresponding to a given value of the effective area ratio A_e in the nonequilibrium solution by the perturbation method. Called by NØNEQ.
AXSECT	For a given list of species pairs, adds those pairs whose cross sections have not previously been specified to a particular step of the transport cross section calculation. Called by XSECT.
BLAYER	Calculates boundary layer momentum and displacement thicknesses, heat flux, and shear stress. Called by FRØZEN, EQUIL, and BICALL.
BICALL	Sets up and executes calls to BLAYER in the non-equilibrium solution. Called by DERIVS and PRTA.

*Throughout this report, the mathematical notation is as defined in Volume I (ref. 1).

BXSECT	Searches a parameter array for references to the indices of species pairs used in the cross section calculation; also determines the correspondence between entries in the parameter array and the steps in the cross section calculation. Called by XSECT.
CØMM	Computes the species production terms $P_i \chi_i$ in the nonequilibrium solution; in runs with an electronic nonequilibrium gas model, also computes the radiative loss and the energy transfer to the electron gas. Called by DERIVS.
CXSECT	Determines the correspondence between species pairs in the master list of species and those in the species list for the current case. Called by XSECT.
DERIVS	Organizes the calculations of rates of change of the species concentrations and other flow variables in the nonequilibrium solution. Called by NØNEQ and RNKT.
DSMSØL	Performs simultaneous solution of a system of linear, inhomogeneous equations by calling subroutine SIMQ. Called by EQCALC, EXACT, NEWRAP, NØNEQ, and PERT.
DUMP	Prints name of routine in which an error has been detected and sets an error indicator to terminate the case. Called by AGSØLN, DSMSØL, EQCALC, FINDX, GEØM, GEØMAR, MATINV, NEWRAP, NØNEQ, PRØP, RESTMP, and WEDGE.
DUMPEA	Prints a large diagnostic dump of common data when a case is terminated by a call to DUMP because of an error. Called by MAIN.
ELCØND	Computes the electrical conductivity of the gas mixture. Called by TRANSP.

ELTIME	Prints elapsed time since the beginning of the run and time since the last previous time message. Called by MAIN and MØDEL.
EPART	Computes parameters for radiative energy loss and energy transfer to the electrons for a specified reaction. Called by CØMM.
EQCALC	Computes mole fractions and thermodynamic conditions for thermochemical equilibrium at specified temperature and pressure. Called by INTA and MØDEL.
(EQUIL)	Entry point in subroutine FRØZEQ; organizes the calculation of the equilibrium flow solution. Called by MAIN.
EXACT	Computes the derivatives of the species concentrations and other flow variables in the non-equilibrium solution. Called by DERIVS.
EXP	In the IBM 360 version, the double precision exponential function e^x with underflow suppression; included to avoid underflow messages. Not used in UNIVAC version. Called by BLAYER, CØMM, EQCALC, NEWRAP, ØUT1, PRØP, RNKT, THERM, and WEDGE.
FINDX	Solves for the axial coordinate x corresponding to a given geometric area ratio A_g in the upstream or downstream region. Called by AGSØLN, AXFIT, FRØZEQ, MAIN, and NEXTMP.
(FINDXC)	Entry point in subroutine FINDX; solves for the axial coordinate x corresponding to a given half width of one of the profiles in a channel. Called by NEXTMP.
(FRØZEN)	Entry point in subroutine FRØZEQ; organizes the calculation of the frozen flow solution. Called by MAIN.

FRØZ EQ	Organizes the frozen and equilibrium flow solutions. The subroutine name is not referenced in NATA; called only through the entry points EQUIL and FRØZEN.
GEØM	Organizes the geometry calculations for the non-equilibrium solution. In the direct integration, it computes A_e and $d \ln A_e / dx$; in the inverse method, it computes ρ and $d \ln \rho / dx$. Called by CØMM.
GEØMAR	Computes the geometric area ratio A_g and its derivative $d A_g / dx$ at a given axial coordinate x . Called by AGSØLN, BLAYER, FINDX, GEØM, ØUT1, and THRØAT.
(GMAR)	Entry point of subroutine GEØMAR; computes the profile ordinate y at a specified axial coordinate x . Called by CØMM, ØUT1, and RADIUS.
(GMAR2)	Entry point of subroutine GEØMAR; computes the profile ordinates y, z at a specified axial coordinate x in a channel. Called by AGSØLN, CØMM, ØUT1, and RADIUS.
(GMAR3)	Entry point of subroutine GEØMAR; computes the profile ordinates, y, z and their derivatives $dy/dx, dz/dx$ at a specified axial coordinate x in a channel. Called by AESØLN, BLAYER, and FINDX.
INGAS	Sets up arrays used in the chemical description of the gas mixture. Called by MAIN.
INIT	Initializes same control parameters and nondimensionalizes the species thermochemical data using the reservoir temperature. Called by MAIN and RESTMP.

(INTA)	Entry point of subroutine INGAS; organizes the calculation of equilibrium conditions in the upstream reservoir for given reservoir temperature and pressure. Called by MAIN and RESTMP.
KANDMU	Calculates quantities proportional to the viscosity and translational thermal conductivity of the gas mixture. Called by TRANSP.
KINT	Computes quantities used in the transport property calculations. Called by TRANSP.
LIST	Prints the "problem summary" portion of the output for each case. Called by MAIN.
MATINV	Computes the inverse A_{ki} of the matrix $\bar{\alpha}_{ij}$ specifying the elemental composition of the independent species. Called by INGAS.
MØDEL	Computes stagnation conditions on models inserted into the flow. Called by FRØZeq, NØNEQ, and PRTA.
NEWRAP	Computes the equilibrium flow conditions at specified temperature and entropy. Called by FRØZeq, MAIN, NØNEQ, and NRMAX.
NEXTMP	Determines the axial coordinate of the next location at which model condition calculations are to be done. Called by FRØZeq and NØNEQ.
NØNEQ	Organizes the nonequilibrium flow solution. Called by MAIN.
NRMAX	Calculates the temperature and other conditions at the sonic point in an equilibrium flow starting at specified upstream reservoir conditions. Called by MAIN and RESTMP.
ØUT	Prints a list of definitions of output variables. Called by MAIN.

ØUT1	Prints conditions in the upstream reservoir and at the sonic points of the frozen and equilibrium solutions. Called by MAIN.
(ØUT2)	Entry point of subroutine ØUT1; prints the conditions at each point in the flow solutions. Called by FRØZEQ and PRTA.
PERT	Computes the perturbations in the flow variables during the nonequilibrium solution by the perturbation method. Called by NØNEQ.
PIØMEG	Computes the Maxwell-averaged electron-neutral momentum transfer cross section for given electron temperature T_e . Called by CØMM.
PRØP	Computes the conditions at a point in the frozen flow solution. Called by FRØZEQ.
PRTA	Organizes the output of conditions at each point in the nonequilibrium flow solution. Called by NØNEQ.
PUTQIN	Organizes the calculations of the cross sections used in computing the transport properties. Called by TRANSP.
QCØUL	Computes the factor $0.8 Q_c$ in the Coulomb cross section for use in the transport property calculations. Called by PUTQIN.
QEX	Computes the cross sections for exchange interaction. Called by PUTQIN.
QEXP	Computes the cross sections for the exponential potential $\phi = Ae^{-r/\rho}$. Called by PUTQIN.
QINTRP	Interpolates tabular data in cross section calculations. Called by QEXP, QLJ, and QTAB.
QLJ	Computes the cross sections for the Lennard-Jones (6-12) potential. Called by PUTQIN.

QMIX	Computes transport cross sections from the mixing rule, equation I(102). [*] Called by PUTQIN.
QREPP	Computes the cross sections for an attractive or repulsive power law potential $\phi = Ar^{-\eta}$. Called by PUTQIN.
QSAME	Sets the cross sections for a species pair equal to a constant multiple of those computed previously for another pair. Called by PUTQIN.
QTAB	Computes the transport cross sections by linear interpolation in tables of $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$, and B^* versus temperature. Called by PUTQIN.
Q11	Computes the transport cross sections using the fairing option, equations I(107) and I(108). Called by PUTQIN.
Q12	Computes the cross sections by the generalized mixing rule, equation I(109). Called by PUTQIN.
Q13	Sets one of the cross sections for a given species pair equal to a constant times another cross section for the same pair. Called by PUTQIN.
Q14	Recomputes the cross sections for a species pair using equations I(112). Called by PUTQIN and Q11.
RADIUS	Computes the profile ordinate, the geometric area ratio, and another quantity required in the calculation of the boundary layer at the first point in the flow solution. Called by BLAYER.

^{*}Equations in Volume I (ref. 1) are cited in the form I().

READ	Reads the input data for each case and sets up arrays defining the flow geometry and gas model. Called by MAIN.
RESET	In the IBM 360 version, simulates a UNIVAC 1108 library routine for storing the time at the start of a run. Called by ELTIME.
RESTMP	Computes the equilibrium conditions in the upstream reservoir based on pressure and mass flow or mass flow and stagnation enthalpy data. Called by MAIN.
RNKT	Computes the changes in the species concentrations and other flow variables over a step in the nonequilibrium integration. Called by NØNEQ.
SHØCK	Computes the pressure ratio for an oblique shock in a perfect gas. Called by WEDGE.
SIMQ	Solves systems of linear inhomogeneous equations. Called by DSMSØL and RESTMP.
STUNTS	Prints an edit of species thermal properties. Called by MAIN.
(STUNT2)	Entry point in subroutine STUNTS; prints edits of the steps in the transport cross section calculation. Called by MAIN.
THERM	Computes the species thermal properties at a specified temperature. Called by DERIVS, EQCALC, FRØZEQ, MØDEL, NEWRAP, and STUNTS.
(THERM1)	Entry point in subroutine THERM; computes only the species specific heats at a specified temperature. Called by MAIN, ØUT1, and TRANSP.
THRØAT	Implements the switch from the upstream solution by the inverse method to the downstream direct integration. Called by NØNEQ.

(TIME) Entry point in subroutine RESET (IBM 360 version only); simulates a library timing routine available on the UNIVAC 1108. Called by ELTIME.

TRANSP Organizes the transport property calculations. Called by BLAYER, FRØZEQ, MAIN, MØDEL, ØUT1, WEDGE, and STUNT2.

(TRANSX) Entry point in subroutine TRANSP; sets up arrays used in the transport property calculations. Called by MAIN.

WEDGE Computes and prints the conditions on the surface of a blunt wedge model inserted into the flow. Called by MØDEL.

WESØLN Calculates the quantities z , zz' , and $(zz')'$, based on the approximate solution I(484) of the wedge equation, for given values of ξ and Γ . Called by WEDGE.

XSECT Sets arrays required for applying the precoded cross section data to the species present in the current gas model and sets up the edited cross section calculation. Called by TRANSX.

2.3 UNIVAC 1108 Version

The UNIVAC version of NATA has been set up to run on the 1108 processors at NASA/JSC under the EXEC II operating system. This version is a single precision program in which some arrays and three subroutines (SHØCK, SIMQ, and WESØLN) are typed as double precision to maintain adequate accuracy. These three routines all perform operations which, under some conditions, can lead to loss of many significant digits because of cancellation of terms. The typing of these routines as double precision is accomplished by retaining the IMPLICIT REAL*8 (A-H, Ø-Z) statements from the IBM 360 version. The UNIVAC Fortran V compiler interprets these statements correctly, a feature included by Sperry Rand in the interest of compatibility with IBM 360 Fortran IV.

NATA is too large to be run on the 1108 without use of overlay. Figure 1 shows the overlay structure employed. The overlay is implemented by means of the EXEC II Memory Allocation Processor (ref. 4). In figure 1, the names enclosed in boxes represent the routines of NATA listed in Section 2.2. Each box represents a segment or subsegment with the associated name (S.---) given in parentheses. The level 0 segment S.MAIN is permanently in core. The various level 1 segments overlay one another. In a typical NATA case, the loading sequence for the level 1 segments is as follows:

S.READ	Read input data, set up gas model, etc.
S.LIST	Print problem summary.
S.INIT	Compute reservoir conditions, etc.
S.XSEC	Set up transport cross section calculation for current gas model.
S.ØUT	Print definitions of output labels.
S.FRØZ	Calculate equilibrium solution to throat, to obtain displacement thickness at throat.
S.INIT	Recalculate reservoir conditions corrected for displacement thickness at throat.
S.FRØZ	Calculate frozen solution; calculate equilibrium solution.
S.NØNE	Calculate nonequilibrium solution.

In the nonequilibrium solution, the subsegment S.N2 is loaded first for use in the initial solution by the perturbation method. When the nonequilibrium integration is started, the integration routine RNKT is loaded, overlaying S.N2.

Control cards for using NATA on the UNIVAC 1108 processors at NASA/JSC are shown in figures 2 and 3. Figure 2

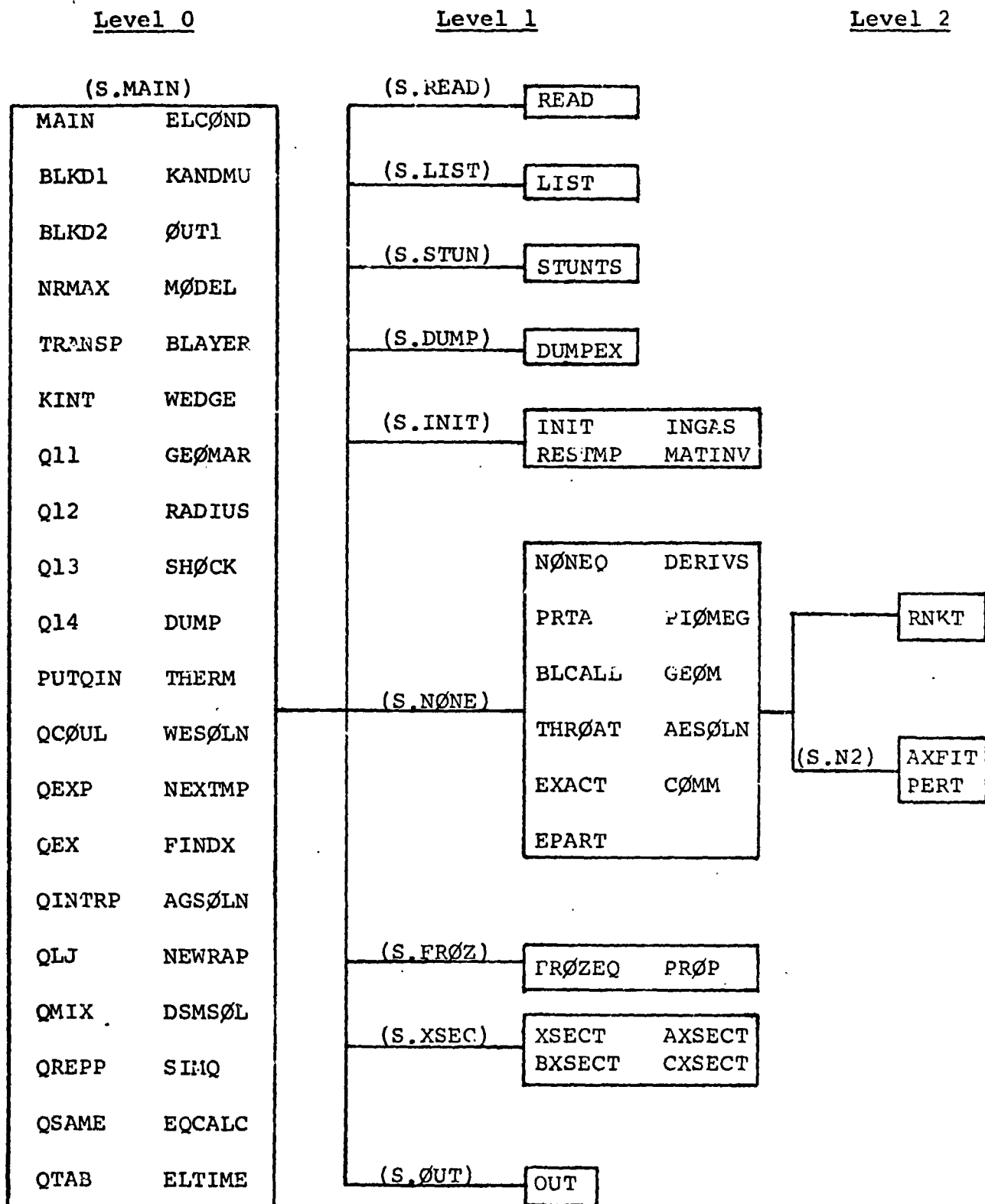


Figure 1. Overlay Structure for UNIVAC 1108 Version of NATA

Z RUN N0320.....

SCOTT/

N MSG FILE REQ. TAPE 01 FH432 0 FSTRN 00

C=OUT

FOR MAIN,MAIN

FOR SUB,SUB

MAP NATA/NATA

S.0 SEG S.MAIN-*(S.READ,S.INIT,S.XSEC,S.FROZ,S.NONE,S.OUT,S.LIST,
S.STUN,S.DUMP)

S.MAIN SEG MAIN-BLKD1-BLKD2-NRMAX-TRANSP-KINT-Q11-Q12-Q13-Q14-PUTQIN-
QCOU-QEXP-QEX-QINTRP-QLJ-QMIX-QREPP-QSAME-QTAB-FLCOND-KANDMU-
OUT1-MODEL-PLAYER-WEDGE-GEOMAR-RADIUS-SHOCK-DUMP-THERM-WESOLN-
NEXTMP-FINDX-A' .OLN-NEWRAP-DSMSOL-SIMQ-EQCALC-ELTIME

S.READ SEG READ

S.LIST SEG LIST

S.STUN SEG STUNTS

S.DUMP SEG DUMPEX

S.INIT SEG INIT-RESTMP-INGAS-MATINV

S.XSEC SEG XSECT-AXSECT-BXSECT-CXSECT

S.FROZ SEG FROZEQ-PROP

S.NONE SEG NONEQ-PRTA-BLCALL-THROAT-EXACT-EPART-DERIVS-PIOMEG-GEOM-AESOLN-
COMM-*(RNKT,S.N2)

S.N2 SEG AXFIT-PERT

S.OUT SEG OUT

XQT CUR

TRW C

DT C

EF C

REL C

Figure 2. Control Cards for Generating a NATA PCF
Tape on the UNIVAC 1108

Z RUN N0320.....

SCOTT/

N MSG FILE REQ. TAPE 01 FH432 0 FSTRN 00

MSG A=V01234

XQT CUR

TRW A

IN A

REL A

XQT NATA

Figure 3. Control Cards for Running NATA on the
UNIVAC 1108

ORIGINAL PAGE IS
OF POOR QUALITY

illustrates a control card deck for generating a NATA PCF tape. The first card is a schematic run card. The deck is set up to compile MAIN and a subroutine "SUB"; the source decks are assumed to be inserted following the "FØR" cards. The "RUN", "MSG", "ASG", "FØR", "MAP", "XQT", "TRW", "ØUT", "TEF", and "REL" cards all have a 7-8 punch in column 1 which was not reproduced by the equipment used to print this listing.

Figure 3 shows the control cards for a NATA run from an existing PCF. The "V01234" following the equals sign on the ASG card is a representative reel number for the PCF tape. The "XQT NATA" card should be followed by the input data for the case, which are described in Section 2 of Volume II (ref. 2).

The control cards required for use of NATA at other UNIVAC 1108 installations may differ in some respects from those illustrated in figures 2 and 3.

2.4 IBM 360 Version

Since NATA is run as a single precision program on the UNIVAC 1108, single precision function names (SQRT, ABS, AMAX1, etc.) are used throughout the program, except in the three double precision routines, SHØCK, SIMØ, and WESØLN. To avoid the tedium of changing function names when the code is converted for use as a double precision program on the IBM 360, the single precision names are retained in the source program for the IBM version and are interpreted as referring to the double precision functions. In the IBM 360/75 installation at Avco Systems Division, this reinterpretation of the single precision Fortran function names is accomplished by using a special subroutine library in the link edit step. At other installations where such a special library may not be available, the same result can be achieved by including, in the NATA source deck, Fortran subroutines defining all of the single precision Fortran function names in terms of the corresponding double precision functions. An example of this procedure is provided by the function EXP in the IBM version of NATA. As shown by the listing of EXP in Section 5, this subroutine redefines EXP in terms of DEXP. It also

sets EXP(X) to 0 for X less than -180, to avoid the production of paper-wasting underflow messages.

Since the Fortran G compiler gives a level 8 diagnostic when a single precision function name is used with a double precision argument, the G compiler cannot be used on the NATA version presented here. Instead, the H compiler, which gives only a level 4 diagnostic, must be used. Several of the NATA routines, including READ and the two block data routines, are too large for the standard H compiler. At Avco Systems Division, an available large compiler (FØRTHMAX) is used to process the NATA source deck.

In both versions of NATA, the elapsed time since the beginning of the run is printed out at several stages in each case. In the UNIVAC version, these times are obtained by calling routines RESET and TIME, which are available in the 1108 system library. In the IBM 360 version in operation at Avco Systems Division, RESET and TIME are simulated using the system library routine ACUCPU, which gives the unexpended CPU time remaining before automatic job termination at TIME.GØ.

In summary, conversion of the UNIVAC version of NATA into the IBM 360 version requires the following operations:

- (1) An IMPLICIT REAL*8 (A-H, Ø-Z) card must be inserted into each NATA routine except ELTIME, ØUT, RESET, SHØCK, SIMØ, and WESØLN. In MAIN, this must be the first card apart from comment cards. In the other routines, the IMPLICIT statement must directly follow the SUBRØUTINE, FUNCTION, or BLØCK DATA statement.
- (2) The subroutines EXP and RESET must be inserted.
- (3) If a special system library for inter, reting the single precision Fortran function names as double precision functions is not available at the computer installation, Fortran subroutines to define the single precision names as double precision functions must be added to the source deck.

Figure 4 illustrates the control cards used to run NATA from a tape containing the binary deck, on the Avco Systems Division IBM 360/75 system. (The required JCL cards will differ somewhat at other IBM 360 installations). The fourth, fifth, and sixth cards in figure 4 invoke the special system library which interprets the single precision function names as referring to the corresponding double precision functions. The binary tape containing NATA is identified on the DD card with VOLUME = SER = NATA3. The data set GØ.FT08F001 is the binary tape upon which NATA can write data for subsequent plotting. GØ.FT07F001 is the data set for punched output.

```

//          JOB                                     .X
//          MSGLEVEL=1
//          XEC FORTHLG,REGION.GO=420K,TIME.GO=4
//LKED.SYSLIB DD DSN=VER1,TEMPLIB,DISP=SHR
// DD DSN=SYS1.FORTLIB,DISP=SHR
// DD DSN=SYS1.DOUBLEP,DISP=SHR
//LKED.SYSUT1 DD SPACE=(1024,(600,50))
//LKED.SYSLMOD DD SPACE=(3072,(200,10,1),RLSE)
//LKED.SYSLIN DD UNIT=(TAPE9.,DEFER),LABEL=(,BLP),DISP=(OLD,KEEP),      X
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),VOLUME=SER=NATA3
// DD *
//          ENTRY MAIN
// *
//GO.FT08F001 DD UNIT=(TAPE9.,DEFER),LABEL=(,BLP),DISP=(NEW,KEEP),      C
//          DCB=(RECFM=VBS,LRECL=92,BLKSIZE=4604),VOLUME=SER=DATASV
//GO.FT07F001 DD SYSOUT=B
//GO.FT05F001 DD *

```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4. Control Cards for Running NATA on the IBM 360

3. ANALYSIS OF ROUTINES

The present section discusses and explains the main program and each subroutine of the NATA code. The purpose of this exposition is to provide an entry into the coding for programmers who wish to analyze errors, make corrections, or introduce modifications into NATA.

All of the physics and gas dynamics and most of the mathematical analysis underlying the programming have already been documented in Volumes I and II of this report (refs. 1, 2). Where appropriate, individual sections and equations of Volumes I and II are referenced.* In cases where the underlying mathematical analysis has not been documented previously, it is explained in the present section.

The routines of NATA are discussed in the following order: first the main program, then all subroutines and functions in alphanumeric order of their names. This is the same order as that in which the routine names are listed in Section 2.2 and at the beginning of Section 5, and in which the source listings are presented in Section 6.

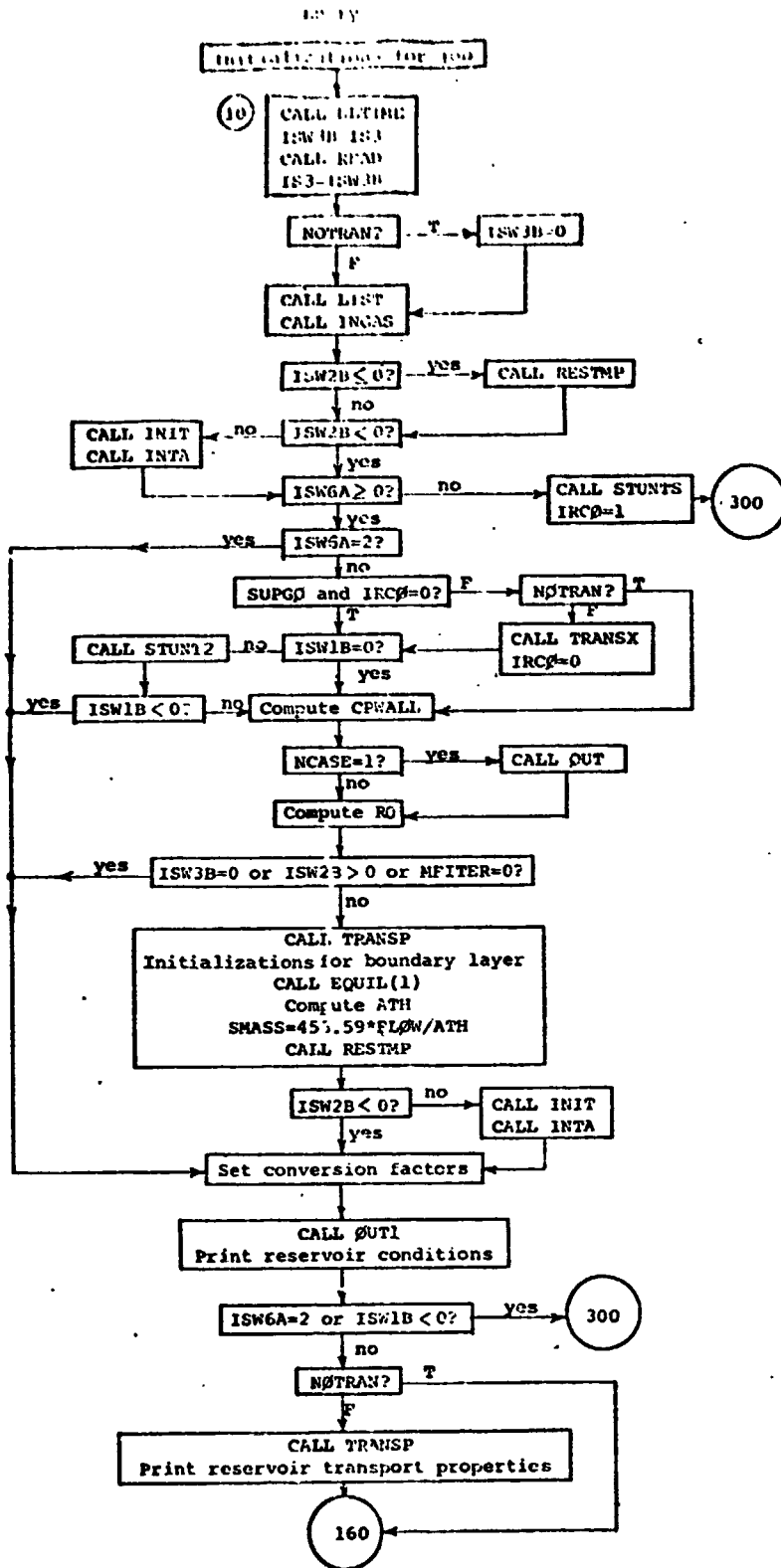
3.1 Main Program

The main program performs certain initialization functions, calls the major subroutines in proper sequence, implements some of the input options, and controls job termination. A flowchart of MAIN is shown in figure 5. The operations shown in this flowchart will be discussed approximately in the order in which they are performed during execution of the code.

3.1.1 Job Initialization

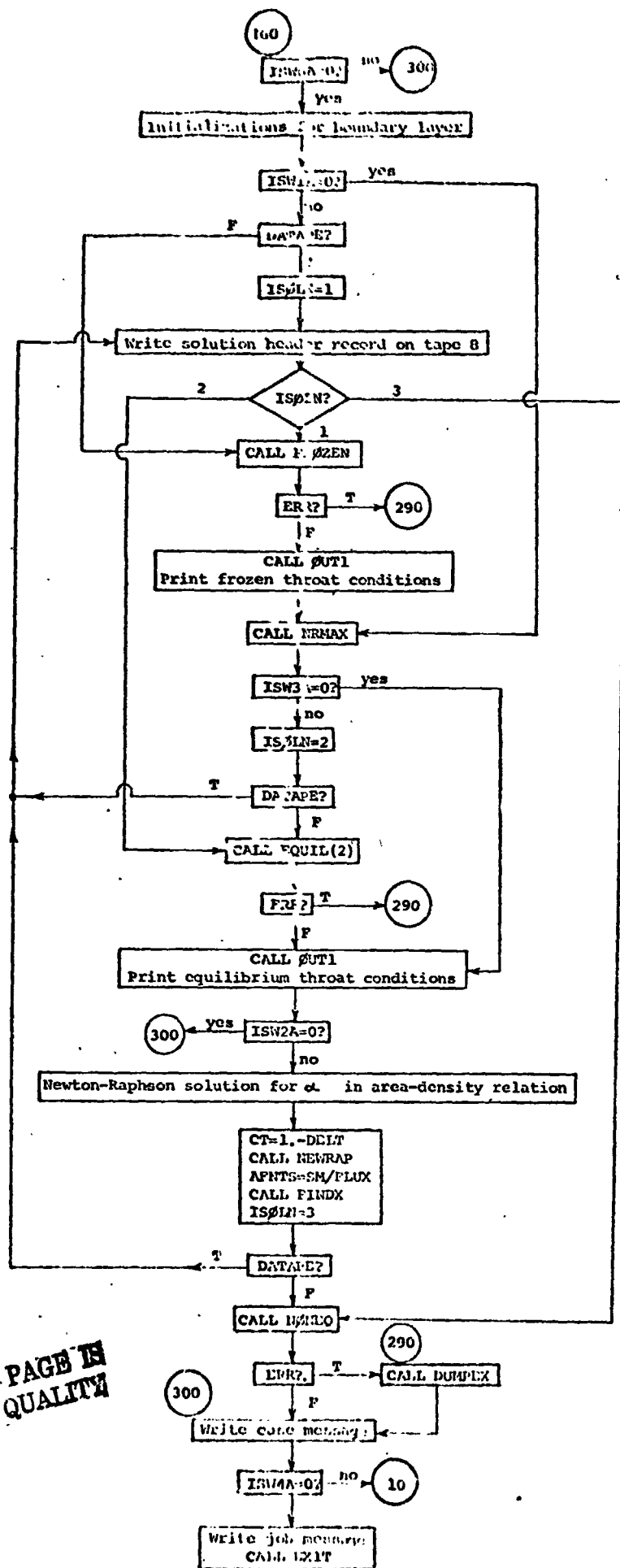
The 12 input control parameters ISW1A to ISW6B are initialized by arithmetic statements in MAIN before any input data

*Equations from Volumes I and II will be referenced in the form I() and II().



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 5a. Flowchart of NATA Main Program (Part a)



ORIGINAL PAGE IS
OF POOR QUALITY

have been read. This procedure is used because these parameters are in unlabelled common, and hence cannot be preset in a BLOCK DATA routine.

3.1.2 Case Initialization

The calculations for each case in a job always begin with statement 10. First, subroutine ELTIME is called to determine the elapsed time ETO since the start of the first case. Then subroutine READ is called. This routine reads the input data for the case, sets up arrays describing the gas model and the nozzle geometry, and prints some of the data defining the case being run. Next, subroutine LIST is called to print the remaining data defining the current case. Subroutine ELTIME is then called again to print the time. ELTIME is called at various locations in MAIN, to provide data on the execution time consumed by major portions of the calculation. These subsequent calls to ELTIME are omitted from the flowchart in the interest of simplification.

Before the call to LIST, the value of the logical input variable NØTRAN is checked. NØTRAN is a switch to allow the suppression of all transport property calculations throughout NATA. With NØTRAN = .TRUE., flow solutions can be run for a gas model containing user-specified species for which transport cross section data have not been provided. Of course, when the transport property calculations are suppressed, the boundary layer cannot be included in the flow solutions. Hence, for NØTRAN = .TRUE., ISW3B is set to zero. However, the input value of ISW3B is saved in the location IS3, and is reset before the call to READ in subsequent cases.

MAIN next increments the case counter NCASE and initializes some variables including the indicator ERR in common block /ERRØR/. This indicator is set to .FALSE. near the beginning of each case. If the programmed checks in any subroutine subsequently detect an error, ERR is set to .TRUE., and this indication causes control to be returned to MAIN as quickly as possible, with no further calculations. Statements in MAIN then cause subroutine DUMPEX to be called. DUMPEX prints out a diagnostic dump consisting of the current values of most of the variables in labelled and unlabelled

common. The details of this scheme are explained more fully in the sections (below) on subroutines DUMP and DUMPEX. Most of the tests on ERR in MAIN have been omitted from the flow-chart for brevity, but the ones following CALL FRØZEN, CALL EQUIL(2), and CALL NØNEQ have been included to illustrate the method.

MAIN next calls subroutine INGAS to set up certain arrays used to describe the chemical nature of the gas mixture. Then, if $ISW2B \leq 0$ (i.e., if the determination of reservoir conditions is to be based partly on the total mass flow), subroutine RESTMP is called to compute the reservoir temperature and pressure. Unless ISW2B is negative, subroutine INIT and INTA are then called. INIT initializes a number of control variables and nondimensionalizes some species properties using the reservoir temperature. INTA computes the species mole fractions and the thermal properties of the gas in the reservoir. For negative ISW2B, INIT and INTA are called in RESTMP with the final reservoir temperature and pressure, and thus do not have to be called in MAIN.

3.1.3 Special Options

If ISW6A is negative, NATA computes and prints tables of thermal properties for all of the species in the current gas model, instead of calculating a flow solution. This operation is performed by subroutine STUNTS.

If ISW6A is positive, NATA computes and prints out the reservoir conditions (as usual), but does not proceed with the flow solutions. For the specific value $ISW6A = 2$, the calculation of the transport properties in the reservoir is also omitted. This feature allows the NATA user to employ the code for thermochemical calculations for a gas model containing nonstandard species without setting up transport cross section data for the species.

If ISW6A is 0 (the normal, preset value), NATA generates one or more flow solutions.

If ISW6A is zero or positive (but not equal to 2), the transport properties in the reservoir are computed. Before transport property calculations can be done, the entry TRANSX of subroutine TRANSP must be called. However, the call to TRANSX can be omitted if TRANSX has been called in a previous case with the same gas model. The gas model is the same as in the preceding case if SUPGØ = .TRUE., and TRANSX was called in a previous case if IRCØ = 0. If NØTRAN is .TRUE., the call to TRANSX is skipped.

Another special option is an edit of the transport cross section calculations, performed by calling the entry STUNT2 of subroutine STUNTS. This call is executed if the input control parameter ISW1B is nonzero. For ISW1B negative, STUNT2 also prints a table of averaged transport cross sections; in this case, no flow solution is run.

3.1.4 Specific Heat of the Gas at the Wall Temperature

MAIN computes the specific heat of the gas at the nozzle wall temperature, TWALL. This calculation is based on the assumption that the gas at this low temperature is a mixture of the "cold species" used to define the elemental composition. The specific heat is calculated as

$$C_{pw} = \frac{R_0}{W_c} \sum_{k=1}^{n_c} x_k \frac{C_{pk}}{R_0} \quad (1)$$

in which W_c is the mean molecular weight of the cold gas, n_c the number of cold species, x_k the mole fraction of the k^{th} cold species, and C_{pk}/R_0 the nondimensional molar specific heat at constant pressure for the k^{th} cold species. This species specific heat is given by CCPJ(I), where $I = \text{IJCS}(K)$ is the index, in the list of species for the current gas model, of the k^{th} cold species.

3.1.5 Definitions of Output Variable Labels

In the first case of each job (NCASE = 1), MAIN calls subroutine ØUT to print a list of definitions, with units,

of the output variable labels used in printing out the flow conditions and test conditions on models. These labels differ, in most cases, from the Fortran variable names used in the code.

3.1.6 Correction of the Reservoir Conditions for the Displacement Thickness at the Throat

If the boundary layer is to be included in the solution ($ISW3B \neq 0$), and if the reservoir condition calculations are based partly on the total mass flow ($ISW2B \leq 0$), then NATA normally recalculates the reservoir conditions, taking account of the effects of the boundary layer displacement thickness on the equilibrium sonic mass flux. For $ISW2B \leq 0$, subroutine RESTMP determines the reservoir temperature and pressure so as to match the sonic mass flux SMASS and one other condition (on pressure or enthalpy). Initially, when RESTMP is called the first time, SMASS is calculated as the ratio of the input total mass flow to the geometric cross sectional area of the nozzle at the throat. To correct these results for the displacement thickness effect, MAIN calls entry EQUIL(IPASS) of subroutine FRØZEQ with the argument IPASS = 1. EQUIL then computes the equilibrium solution up to the throat, including the boundary layer. MAIN then calculates the effective cross sectional area ATH at the throat including the effects of the displacement thickness, recomputes the sonic flux SMASS, and calls RESTMP for a second time to recompute the reservoir conditions. This correction can be suppressed, if it is not desired, by setting MFITER = 0 in the input.

3.1.7 Conversion Factors

Next, MAIN sets the array of conversion factors CF(I) in common block /CØNVRT/. These factors convert the non-dimensional flow variables used in the internal computations to the units used in the printed output.

3.1.8 Output of Reservoir Conditions

The nondimensional flow variables $CT = T/T_0$, $PRES = p/p_0$, etc., are now set to their reservoir values and subroutine ØUT1 is called to print out the reservoir conditions.

3.1.9 Solution Header Records on Tape 8

If the frozen flow solution is to be generated ($ISW1A \neq 0$) and if data are to be written on a binary output tape for subsequent plotting ($DATAPE = .TRUE.$), the main program next writes a "header" record on tape 8 for the frozen solution. This record contains the nozzle index $NØZZLE = NPRØFL(1)$, the type of solution ($ISØLN = 1$ for frozen flow), and the reservoir conditions.

3.1.10 Frozen Solution

To generate the frozen flow solution, MAIN calls the entry $FRØZEN$ of subroutine $FRØZeq$. It then calls $ØUT1$ to print the sonic-point conditions in the frozen solution.

3.1.11 Equilibrium Solution

MAIN calls subroutine $NRMAX$ to compute the conditions at the sonic point in the equilibrium solution. These data are needed for both the equilibrium and nonequilibrium solutions. Then, unless $ISW3A = 0$, MAIN checks $DATAPE$. If this control variable is $.TRUE.$, a header record is written on tape 8 for the equilibrium solution, and the entry $EQUIL(IPASS)$ of subroutine $FRØZeq$ is called with $IPASS = 2$ to produce the solution. Finally, $ØUT1$ is called to print out the equilibrium throat conditions.

3.1.12 Nonequilibrium Solution

If the nonequilibrium solution is to be computed ($ISW2A \neq 0$), MAIN determines the constants C and α in the area-density relation $I(383)$ assumed in the solution by the inverse method upstream of the throat. The parameter α is determined by a Newton-Raphson solution of equation $I(385)$ or $I(386)$. The iteration is started using the initial estimate

$$\alpha \approx 4 + 8 \ln \left(\frac{\rho}{\rho_0} \right) \quad (2)$$

which can be derived by taking the logarithm of $I(385)$ and

expanding $\ln (\alpha + 2)$ around $\alpha = 0$:

$$\ln (\alpha + 2) \simeq 2 + \frac{\alpha}{2} - \frac{\alpha^2}{8} \quad (3)$$

Then to obtain starting conditions for the perturbation solution, MAIN sets the nondimensional temperature CT to 1-DELT, calls subroutine NEWRAP to compute the equilibrium flow conditions at this temperature, computes the corresponding effective area ratio AFNTS, and calls subroutine FINDX to determine the axial coordinate CX at which these conditions occur. If DATAPE = .TRUE., a header record is written on tape 8 for the nonequilibrium solution. Then subroutine NØNEQ is called to generate the nonequilibrium flow solution.

3.1.13 Diagnostic Dumps

If a .TRUE. value of the error indicator ERR is detected anywhere in MAIN, control is immediately transferred to the statement CALL DUMPEX, which prints out the current values of a large number of common variables as an aid to identifying the cause of failure. If data have been written on tape 8, the tape is backspaced to eliminate the records written during the failed case.

3.1.14 Job Termination

At the end of the calculations for each case, a line of output is written containing the case number, the number of records written on tape 8 (if any), and the total number of records written thus far in the job. Then the value of the job termination control parameter ISW4A is checked. If ISW4A is nonzero, the input data for the next case are read. If ISW4A is zero, a summary of the completed and failed cases in the job is written and, if DATAPE is .TRUE. an end file mark is written on tape 8 and the tape is rewound.

3.2 Subroutine AESØLN

This subroutine calculates the effective area ratio at a given position, X, in the nozzle. The value X is a subroutine

argument. Other input data to the routine are contained in the common block /AEGEØM/, namely, the geometric area ratio S1, the derivative of the geometric area ratio S2, and the square root of the geometric area ratio SQRTA. To set S1 and S2, a call to AESØLN(X) must be preceded by a call to GEØMAR(X,S1,S2). In addition, if NPRFLS = 1 (the flow is in a nozzle, not a channel) and JDIM = 1 (the nozzle is axisymmetric), then SQRTA must be computed in the calling routine before AESØLN is called.

The calculations of the effective area ratio are based upon equations I(126), I(130) and I(134), respectively, for the cases of a two-dimensional nozzle (NPRFLS = 1, JDIM = 0), an axisymmetric nozzle (NPRFLS = 1, JDIM = 1), and a rectangular channel (NPRFLS = 2). The effective area ratio obtained is stored in the location S1 of common block /AEGEØM/, replacing the geometric area ratio.

In addition, the subroutine computes the derivative, dA_e/dx , of the effective area ratio, using the following formulas:

2D Nozzle

$$\frac{dA_e}{dx} = \frac{\frac{dA_g}{dx} - \frac{1}{y_0} \frac{d\delta^*}{dx}}{1 - \delta^* / y_0} \quad (4)$$

Axisymmetric Nozzle

$$\frac{dA_e}{dx} = \sqrt{A_e} \left[\frac{\frac{1}{\sqrt{A_g}} \frac{dA_g}{dx} - \frac{2}{y_0} \frac{d\delta^*}{dx}}{1 - \delta^* / y_0} \right] \quad (5)$$

Channel

$$\frac{dA_e}{dx} = \frac{(z - \delta_2^*) \left(\frac{dy}{dx} - \frac{d\delta_1^*}{dx} \right) + (y - \delta_1^*) \left(\frac{dz}{dx} - \frac{d\delta_2^*}{dx} \right)}{(y_0 - \delta_{1*}^*) (z_0 - \delta_{2*}^*)} \quad (6)$$

These relations can easily be derived from equations I(126), I(130), and I(134). The computed value of dA_e/dx is stored in the location S2 of common /AEGEØM/, where it replaces the derivative of the geometric area ratio.

3.3 Subroutine AGSØLN

The purpose of this subroutine is to solve for the geometric area ratio A_g when the effective area ratio A_e and boundary layer displacement thickness δ^* are given. These input data are provided in the argument list: $A_e = AE$, $\delta_i^* = DEL(i)$, where the index i runs from 1 to 2 in the case of a channel. The argument list also includes the indicator UPDØWN, which specifies whether an upstream (-1.) or downstream (+1.) solution is desired. The computed value of A_g , and the corresponding value of x , are returned to the calling routine through the arguments AG and X, respectively.

The solution for A_g is straightforward and explicit in the case of a nozzle. It is given by equation I(127) if the flow geometry is two-dimensional and by I(131) if it is axisymmetric. In both cases, the value of X corresponding to A_g is determined by calling FINDX.

In the case of a channel, the relation between A_e and A_g is given by equation I(135) and the profile functions $y(x)$, $z(x)$. This relation does not permit an explicit solution for A_g . In AGSØLN, a numerical Newton-Raphson method is used to determine A_g . Equation I(135) is written as

$$A_g = A_c + \frac{y \delta_2^* + z \delta_1^*}{y_0 z_0} \quad (7)$$

where

$$A_c = \left(1 - \frac{\delta_1^*}{y_0}\right) \left(1 - \frac{\delta_2^*}{z_0}\right) A_e - \frac{\delta_1^* \delta_2^*}{y_0 z_0} \quad (8)$$

The quantity A_c and the factors $\delta_1^*/y_0 z_0$ and $\delta_2^*/y_0 z_0$ are constants during the solution for a particular call to the subroutine. From (7), the object of the calculation is to find a value of x such that $F(x) = 0$, where

$$F(x) \equiv A_c + \frac{y(x) \delta_2^* + z(x) \delta_1^*}{y_0 z_0} - A_g(x) \quad (9)$$

Let x , F denote the values of these quantities from the current iteration and x_0 , F_0 the values for the previous iteration. Then, based on a linear approximation to (9), the value of x required to make $F(x) = 0$ is estimated to be

$$x_n = x - F \cdot \frac{x - x_0}{F - F_0} \quad (10)$$

The iteration based on this relation begins at statement 60 in the subroutine. To start the iteration, A_g is set equal to A_e , and the corresponding x is computed using FINDX. The second point is obtained by computing A_g from (7) and again using FINDX. Equation (10) is then used in the third and all subsequent steps. The convergence criterion is that $|F|/A_g$ be less than or equal to 10^{-5} , or that $F = F_0$.

If $A_g(x)$ in (9) is expressed by means of equation I(120), equation (9) can be rewritten in the form

$$F(x) = \left(1 - \frac{\delta_1^*}{y_0}\right) \left(1 - \frac{\delta_2^*}{z_0}\right) A_e - \left(\frac{y}{y_0} - \frac{\delta_1^*}{y_0}\right) \left(\frac{z}{z_0} - \frac{\delta_2^*}{z_0}\right) \quad (11)$$

If y and z both have minima at the throat $x = 0$, as is normally the case in channels, then the maximum value of $F(x)$ occurs at $x = 0$ and is equal to

$$F_{\max} = F(0) = \left(1 - \frac{\delta_1^*}{y_0}\right) \left(1 - \frac{\delta_2^*}{z_0}\right) A_e - \left(1 - \frac{\delta_1^*}{y_0}\right) \left(1 - \frac{\delta_2^*}{z_0}\right) \quad (12)$$

In order for a solution of the equation $F(x) = 0$ to exist, it is necessary that F_{\max} be greater than or equal to zero. This is always true if $\delta_1^* = \delta_{1*}^*$ and $\delta_2^* = \delta_{2*}^*$. Also, in practice, it is always true when $A_e \gg 1$. However, when A_e is nearly equal to 1 and $\delta_1^* < \delta_{1*}^*$, $\delta_2^* < \delta_{2*}^*$, F_{\max} can be negative. These circumstances can arise near the sonic point (where $A_e = 1$) when the sonic point is displaced from the geometric throat by the presence of large gradients in the displacement thicknesses. When this occurs, the iteration formula (10) extrapolates past the throat and gives an x_n of the wrong sign, i.e., the opposite sign to UPDOWN. When such a condition is detected, AGSOLN attempts to "fix" the problem by resetting the displacement thicknesses δ_1^* , δ_2^* to their values at the sonic point, δ_{1*}^* and δ_{2*}^* respectively, and restarting the iterative solution at the beginning. Under the circumstances in question, this adjustment can have little effect on the value of A_g computed (which is near unity). Equation (12) shows that after this replacement, F_{\max} is the product of a positive factor and $(A_e - 1)$, and hence is ≥ 0 . Thus, if the problem was indeed caused by the combination of circumstances described above, the "fix" should guarantee the existence of a solution of the equation $F(x) = 0$. The coding provides a dump and an error exit should the problem recur after the "fix" has been carried out. In this case, the problem has a different cause, such as an error in a user-supplied channel profile curvefit.

3.4 Subroutine AXFIT

This subroutine solves for the value of x corresponding to a given effective area ratio (AFNTS) during the initial (perturbation) portion of the nonequilibrium solution. If the solution is being calculated without the boundary layer (ISW3B = 0), then the geometric area ratio is equal to the effective area ratio, and x (denoted by CX) is determined by calling subroutine FINDX. If the boundary layer is included in the solution, but subroutine THROAT has not yet been called to reset IUPD from 1 to 0, the effects of the boundary layer displacement thickness on the effective flow geometry are neglected, and x is again obtained from FINDX. However, if the boundary layer is included and IUPD = 0, then the inviscid flow is coupled to the boundary layer and

FINAL PAGE IS
OF POOR QUALITY

x is accordingly determined by calling AGSØLN. Both FINDY and AGSØLN require an indicator which specifies whether the upstream (UPDØWN = -1) or downstream (UPDØWN = 1) solution for the x corresponding to the effective area ratio AFNTS is desired. In AXFIT, this indicator is determined by testing whether the nondimensional temperature CT is greater than or less than its value CTMAX at the geometric throat based on the equilibrium flow solution.

3.5 Subroutine AXSECT

Subroutine AXSECT is called by subroutine XSECT for the purpose of adding data for additional species pairs to the edited cross section data in common block/TRANS7/. The first argument LL of the subroutine gives the step in the cross section computations at which the additional pairs are to be added, while the second argument NN gives the total number of species to be considered in adding the data. The indices of the actual species to be considered in the added data are then given, in numerical order, by the first NN entries of the variable I in common block/TRANS2/. Subroutine AXSECT searches through all pairs (II, JJ) of these species with $II \leq JJ$, to find those for which cross section data have not been previously specified, as indicated by the fact that $Q(1, II, JJ) = 0$. All such pairs are then added to the IQ, JQ arrays, starting at the location $NQ(LL) + 1$, with the original data in these locations being moved back to make room for the new values. Further, the subroutine sets $Q(1, II, JJ) = 1$. for all pairs added to the IQ, JQ arrays and revises the NQ array to be consistent with the new IQ, JQ values, assuming that all of the added pairs are to be included in the LL^{th} step of the cross section computations.

3.6 Subroutine BLAYER

The boundary layer calculations in NATA are carried out by the method described in Section 5 of Volume I (ref. 1). These calculations are performed by subroutine BLAYER. The boundary layer properties are calculated from the integral I, equation I(172). This integral is used to determine the correlation parameter n, equation I(174). The momentum thickness

θ , displacement thickness δ^* , shear stress τ_w , and heat flux q_w are then calculated from n and other quantities.

The overall structure of BLAYER is diagrammed in figure 6. The calculations are all contained in a DØ loop with index L running from 1 to NPRFLS. In the case of a nozzle (NPRFLS=1), the calculations are executed only once. For a channel (NPRFLS=2), they are executed twice to compute the boundary layers on the two pairs of channel walls.

The subroutine argument FINAL is a logical variable specifying whether the new flow point is an actual point of the solution or merely an intermediate step in the Runge-Kutta integration used to generate the nonequilibrium flow solution. For FINAL = .FALSE., the important variables used in BLAYER are restored to their values at the beginning of the step before the RETURN is executed, and the calculations of shear stress TAUW and heat flux QWDØT are omitted.

IPØINT is a counter for the flow points at which boundary layer calculations are done in each flow solution. IPØINT is initialized to 1 in subroutine FRØZEQ for frozen and equilibrium solutions and in subroutine NØNEQ for nonequilibrium solutions. Each time BLAYER is called with FINAL = .TRUE., IPØINT is incremented by 1 before the RETURN.

Some of the physical variables which are used frequently are given short mnemonic names which are equivalenced to their standard names in unlabelled common

3.6.1 Initializations

When BLAYER is called for the first time in a given flow solution (IPØINT = 1), a number of quantities are set or initialized by the statements down to 90. These quantities include the coefficient B in the linear relation $I(164)$ between the momentum parameter N and the correlation parameter n , and the transport properties $PRW = Pr_w$ and $VISCW = \mu_w$ at the wall temperature. The wall is assumed to be catalytic; hence, these transport property calculations are based on the mole fractions QPJ(K) for the cold gas mixture. This section of the routine also sets the indicator

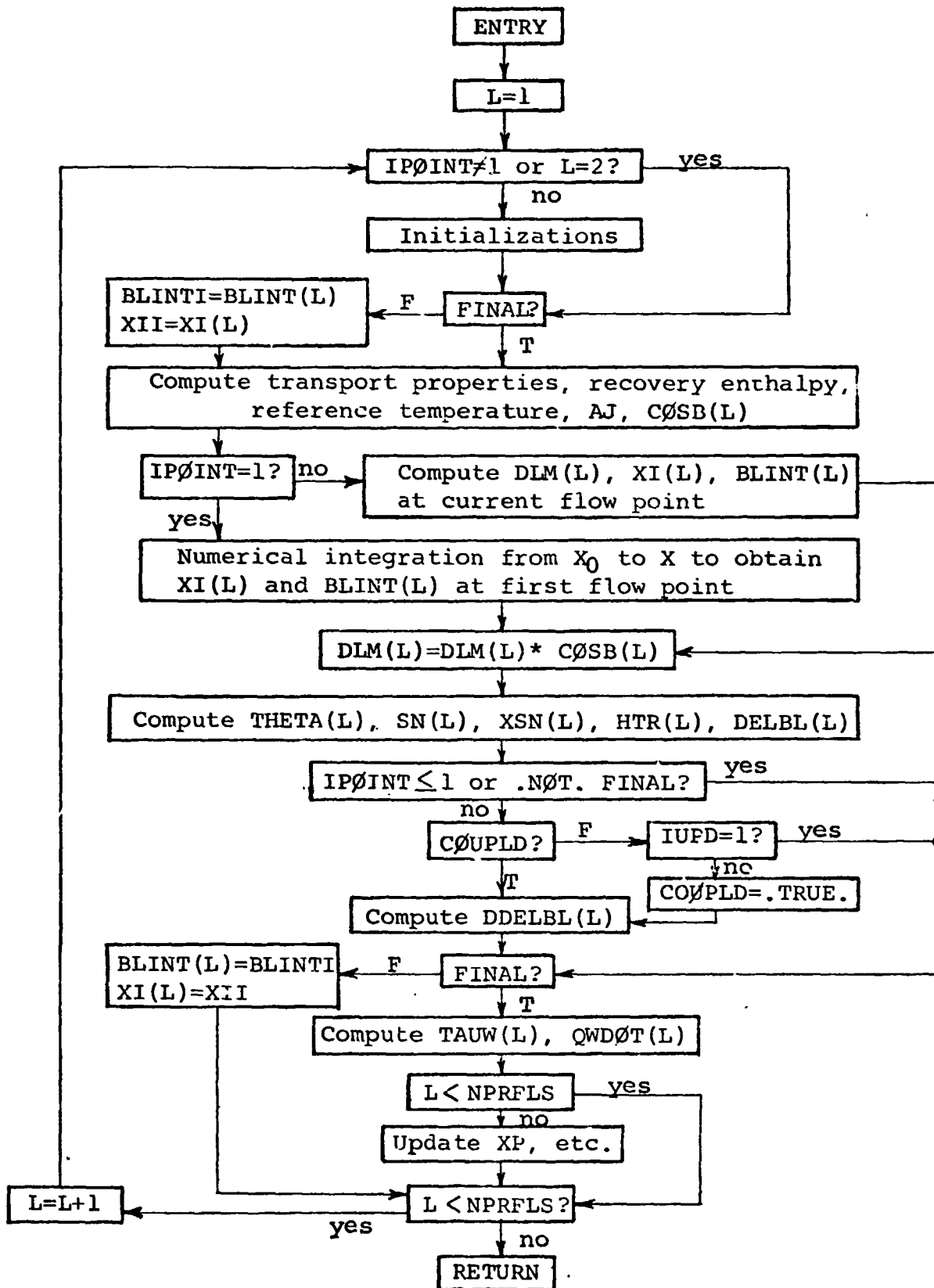


Figure 6. Flow Chart of Subroutine BLAYER

ITYPE, which specifies the type of nozzle geometry:

ITYPE = 1 Two-dimensional nozzle

ITYPE = 2 Axisymmetric nozzle

ITYPE = 3 Rectangular channel

3.6.2 Transport Property and Geometry Calculations

The section of BLAYER from statement 90 down to 270 computes the values of various physical quantities at the current flow point, including the transport properties and geometric parameters entering into the boundary layer calculation.

BLAYER uses the viscosity and Prandtl number at three temperatures, the wall temperature TWALL, the free-stream temperature, TE, and the reference temperature TREF. The calculations for TWALL are done only once per solution because the wall temperature is assumed constant. To save computing time, the transport property calculations at TE and TREF are not repeated at every flow point, but are done only when the temperature or the gas mole fractions have changed significantly. The criterion for performance of the transport property calculations is a change of 0.01 in any of the mole fractions or a change of 0.5 percent in the temperature. The mole fractions at which calculations were last done are represented by XTEST(J). The corresponding temperatures are TEP and TREFP.

The factor r^{2j} in equation I(171) is denoted by AJ in BLAYER. Because of the form of equations I(171) to I(173), r^{2j} can be replaced by a constant times r^{2j} without affecting the computed results for the boundary layer thicknesses, the heat flux, and the shear stress. For two-dimensional nozzles, AJ is set to 1, which is the value of r^{2j} for $j = 0$. For axisymmetric nozzles, AJ is set to the geometric area ratio,

$$A_g = (y/y_0)^2 \quad (13)$$

which is r^2_j/y_0 . In the case of a rectangular channel, for each profile, AJ is set to the square of the radius of the other profile, since this radius is proportional to the width of the surface whose boundary layer is being calculated.

For nozzles, the cosine of the angle b between the profile tangent and the axis is calculated from dA_g/dx . In general,

$$\cos b = \frac{1}{\sqrt{1 + \tan^2 b}} = \frac{1}{\sqrt{1 + (dy/dx)^2}} \quad (14)$$

where y is the profile ordinate. In the case of a two-dimensional nozzle, from equation I(118),

$$\frac{dy}{dx} = y_0 \frac{dA_g}{dx} \quad (15)$$

and hence

$$\cos b = \frac{1}{\sqrt{1 + y_0^2 (dA_g/dx)^2}} \quad (16)$$

In the axisymmetric case, from I(116),

$$\frac{dy}{dx} = \frac{y_0}{2\sqrt{A_g}} \frac{dA_g}{dx} \quad (17)$$

and thus

$$\cos b = \frac{1}{\sqrt{1 + \frac{y_0^2}{4 A_g} \left(\frac{dA_g}{dx} \right)^2}} \quad (18)$$

The derivative dA_g/dx in (16) and (18) is obtained by calling subroutine GEØMAR. In the case of a channel, dy/dx is computed by calling the entry GMAR3 of GEØMAR, and $\cos b$ is evaluated directly from (14).

The viscosity-temperature exponent ω , defined by equation I(191), is determined by fitting I(191) to the viscosity at the wall and reference temperatures. The coefficient A in I(164) (denoted by EK1) is calculated from I(201) at the first flow point (IPØINT = 1), and then held constant through the solution.

The integrand \mathfrak{F} of I(172) is represented by ØRDIN(L). Actually, ØRDIN(L) differs from equation I(171) by a constant factor, which cancels out in the calculation of n' I(173) and thus has no effect on the boundary layer properties.

3.6.3 Integration from x_0 to First Flow Point

In general, the flow point at which BLAYER is first called lies upstream of the throat. The Mach number at this point is low. However, the point may be well downstream of the point x_0 at which boundary layer development is assumed to start (see Sec. 5.10 of Volume I). Accordingly, the interval from x_0 to the first point of the flow solution is not treated as a single step for the purpose of the boundary layer calculation. Instead, when BLAYER is called with IPØINT = 1, the integral I(172) (denoted by BLINT(L)) and the streamwise boundary layer coordinate \mathfrak{F} (denoted by XI(L)) are evaluated by a 10-step numerical integration from x_0 to the first flow point.

In the region covered by this integration, the Mach number is low so that the flow can be approximated as incompressible. Also, the boundary layer is thin in this region, in comparison with the nozzle or channel radius, so that the effective area ratio can be approximated by the geometric area ratio. Thus, the continuity equation $\rho u A_e = \text{constant}$, can be approximated

$$M A_g = \text{constant} \quad (19)$$

The constant in (19) is represented by CC. At each point in the integration from x_0 to x_1 , the geometric area ratio $A_g = A_{g1}$ and the quantity AGJ corresponding to r^2J are calculated by calling subroutine RADIUS, M is obtained from (19), and a quantity ØRD1 proportional to ØRDIN(L) is computed with the

approximation $a_e/a_0 \approx p_e/p_0 \approx 1$. The increment in ξ is computed using the Pythagorean theorem:

$$(\Delta \xi)^2 = (\Delta x)^2 + (\Delta y)^2 \quad (20)$$

where Δy is the change in the profile ordinate due to a change Δx in the axial coordinate. BLINT(L) is calculated using trapezoidal-rule integration. At the first flow point, DLM(L) = $d \ln M/dx$ is evaluated from (19).

3.6.4 Derivatives of the Mach Number

The derivative dM_e/dx of the flow Mach number is an important variable in the boundary layer calculation because it represents the effect of a streamwise pressure gradient. The related quantity $dM_e/d\xi$ appears in the formula I(174) for the correlation parameter, n . All of the boundary layer properties depend upon n .

The derivative $dM_e/d\xi$ differs between the two profiles of a channel because the boundary layer coordinate lies along the channel surface and the two profiles have different inclinations to the nozzle axis. The value of $d \ln M_e/d\xi$ for the L^{th} profile is denoted by DLM(L). It is determined by first computing $d \ln M_e/dx$ and then correcting for the inclination of the nozzle or channel surface to the axis. For each flow point beyond the first, $d \ln M_e/dx$ is calculated from rate data if the flow solution is of non-equilibrium type and the numerical integration technique is being used (INEQ = 1). Otherwise, if the flow solution is of frozen or equilibrium type or if the perturbation technique is being used in a nonequilibrium solution (INEQ = 0), $d \ln M_e/dx$ is calculated by numerical differentiation. If the inviscid solution is being calculated by nonequilibrium integration, DLM(L) is calculated from equations I(420), I(421), and I(291) or I(422). In the case of a chemical nonequilibrium gas model (NT = 1), the stagnation enthalpy h_0 is constant and hence I(421) and I(291) give

$$\frac{du}{dx} = -\frac{1}{u} \sum_{j=1}^n \left[\gamma_j \frac{dH_j}{dx} + H_j \frac{d\gamma_j}{dx} \right] \quad (21)$$

The variables used in the code are

$$SHJ(j) = \frac{H_j}{R_c T_0} \quad (22a)$$

$$CCPJ(j) \cdot DT = \frac{1}{R_0 T_0} c_{pj} \frac{dT}{dx} = \frac{1}{R_0 T_0} \frac{dH_j}{dx} \quad (22b)$$

$$SU = u \sqrt{\frac{W_0}{R_0 T_0}} \quad (22c)$$

$$GJ(j) = \gamma_j \quad (22d)$$

Hence (21) is coded as

$$\begin{aligned} DUDX \equiv \frac{d(SU)}{dx} &= - \frac{W_0}{SU} \sum_{j=1}^n [GJ(j) \cdot CCPJ(j) \cdot DT + SHJ(j) \cdot DGJ(j)] \\ &= - \frac{CMA}{SU} (SCPG \cdot DT + SHJDGJ) \end{aligned} \quad (23)$$

The sums denoted by SCPG and SHJDGJ are obtained from common. Similarly, in the case of an electronic nonequilibrium model (NT = 2), equations I(421) and I(422) give

$$\begin{aligned} DUDX &= \frac{DCHA}{SU} - \frac{CMA}{SU} [GJ(1) \cdot CCPJ(1) \cdot DTE \\ &\quad + SCPGH \cdot DT + SHJDGJ] \end{aligned} \quad (24)$$

where

$$DCHA = \frac{W_0}{R_c T_0} \frac{dh_0}{dx} \quad (25)$$

$$SCPGH = \sum_{j=2}^n GJ(j) \cdot CCPJ(j) \quad (26)$$

In (24), quantities subscripted (1) refer to the electron species.

For equilibrium and frozen solutions, and for nonequilibrium solutions which are being developed by the perturbed equilibrium flow technique (INEQ = 0), the quantities SCPG and SHJDGJ are not available. In these cases, DLM(L) is calculated by numerical differentiation. For IPØINT = 2, a simple first-order difference expression

$$\frac{dM}{dx} = \frac{M-M_p}{\Delta x} \quad (27)$$

is used, where M is the Mach number at the current point and M_p that at the previous point. For IPØINT ≥ 3, the following second-order difference calculation is used: Let (M,x) be the data for the current point (M_p, x_p) those for the previous point, and (M_{pp}, x_{pp}) those for the point preceding the previous one. A quadratic fit to these data can be written in the form

$$M = c_0 + c_1x + c_2x^2 \quad (28)$$

from which

$$\frac{dM}{dx} = c_1 + 2c_2x \quad (29)$$

Substitution of the three data points into (28) gives, after elimination of c_0 ,

$$\frac{M-M_p}{x-x_p} = c_1 + c_2 (x + x_p) \quad (30a)$$

$$\frac{M-M_{pp}}{x-x_{pp}} = c_1 + c_2 (x + x_{pp}) \quad (30b)$$

Hence

$$c_2 = \frac{\frac{M-M_p}{x-x_p} - \frac{M-M_{pp}}{x-x_{pp}}}{x_p - x_{pp}} \quad (31)$$

Now that c_2 is known, c_1 can be obtained from (30b);

$$c_1 = \frac{M-M_p}{x-x_p} - c_2 (x + x_p) \quad (32)$$

dM/dx is then calculated from (29), and $DLM(L)$ is obtained from

$$\frac{d \ln M}{dx} = \frac{1}{M} \frac{dM}{dx} \quad (33)$$

3.6.5 Calculation of Boundary Layer Thicknesses

Next, the increase in ξ in the step from the previous flow point to the current one is calculated,

$$\Delta \xi = \frac{\Delta x}{\cos b} \quad (34)$$

and ξ is incremented by $\Delta \xi$. The boundary layer integral I (equation I(172)), denoted by $BLINT(L)$, is updated using the trapezoidal integration rule. The derivative $d \ln M/dx$ is converted into $d \ln M/d\xi$ by

$$\frac{d \ln M}{d\xi} = \cos b \frac{d \ln M}{dx} \quad (35)$$

Then n' , equation I(173), is evaluated, and the correlation parameter n is calculated from I(174). The characteristic length L in these equations is taken to be R_0 , defined as the profile radius in the case of a nozzle, and defined by

$$\pi (R_0)^2 = \text{throat area} \quad (36)$$

in the case of a channel. The momentum thickness is calculated from I(184). If the flow geometry is that of an axisymmetric nozzle, the transverse curvature correction I(187) is applied. The smoothed correlation parameter \bar{n} , denoted by XSA(L), is computed from equation I(217).

The effect of free stream Mach number upon the correlations of boundary layer properties is contained in the dependence upon the hypersonic parameter σ , equation I(190). The correlations used in BLAYER are based upon the Dewey-Gross calculations (ref. 5) for a perfect gas. In frozen and nonequilibrium solutions, the parameter σ as calculated from I(190) never approaches unity, even at high Mach numbers, because of the frozen-in dissociation energy in h_0 . To represent the Mach number effect more realistically, the σ defined by I(190) is replaced by an effective value σ_e , defined as

$$\sigma_e = \left[1 + \frac{2}{(\gamma_e - 1) M^2} \right]^{-1} \quad (37)$$

where γ_e is an effective specific heat ratio. For a perfect gas, (37) is equivalent to I(190). In BLAYER, γ_e is calculated as

$$\gamma_e = a^2 W / R_0 T \quad (38)$$

where W is the mixture molecular weight and a the sound speed.

The incompressible form factor HTR(I) is computed from I(189) and the displacement thickness DELBL(L) from I(188). In the case of a nonequilibrium flow solution, the coupling of the boundary layer to the inviscid flow is switched on when the indicator IUPD is set to zero in subroutine THRAT.* The only effect of the coupling in subroutine BLAYER is that the derivatives DDELBL(L) of the displacement thicknesses

*IUPD is always equal to 1 in the frozen and equilibrium solutions.

have to be calculated. To avoid excessive disturbance of the nonequilibrium solution, DDELBL(L) is gradually built up from zero over 29 steps in the flow solution, as explained in Section 7.6 of Volume I.

3.6.6 Heat Flux and Shear Stress

For FINAL = .FALSE., the boundary layer integral BLINT(L) is reset to its value BLINTI at the beginning of the step, and the boundary layer coordinate XI(L) to its initial value XII. Then control is transferred to statement 540, the limit of the DØ loop over L.

For FINAL = .TRUE., the shear parameter XL is calculated from the curvefit I(194), the shear stress TAUW(L) from I(193), the Reynolds analogy factor RAF from I(200), and the heat flux QWDØT(L) from I(195). Then, if L is less than NPRFLS, control is transferred to the end of the DØ loop over L. If L is equal to NPRFLS, several physical quantities are saved for use in the calculations for the next step. When the DØ loop over L has been completed, diagnostic outputs are written if the control variable ISW4B is nonzero, and the RETURN is executed.

3.7 Subroutine BLCALL

During the nonequilibrium solution, all calls to the boundary layer routine BLAYER are made through subroutine BLCALL. Before calling BLAYER, subroutine BLCALL sets up certain data required by BLAYER and by the transport property routines. The species mole fractions are computed from equation I(1) and loaded into the common array SAVEC(J). If the perturbation technique is still being used (INEQ = 0), the perturbations PERTGJ(J) are added to the concentrations $\gamma_j = GJ(J)$ before equation I(1) is used. In this case, in addition, the temperature perturbation PCT is added to the nondimensional temperature CT and the density perturbation PRHØ is added to RHØ. After BLAYER has been called, the temperature and density are restored to their unperturbed (equilibrium) values. It is not necessary to reset SAVEC(J), because this array is used only in the transport, boundary layer, and model condition calculations, not in the inviscid

flow solution. Indeed, subroutine PRTA (which controls the output of the nonequilibrium flow solution) calls BLCALL to load the SAVEC array even in solutions neglecting the boundary layer (ISW3B = 0), because the output routine ØUT1 obtains the mole fractions from SAVEC.

3.8 Subroutine BXSECT

Subroutine BXSECT is used in searching the array V in common block/TRANS7/for references to the indices of species pairs used in the cross section computations, and also in determining the correspondence between the entries in the V array and the steps in the cross section computations. The argument L of the subroutine is the index of the step in the cross section computations, MV is the index of the last entry in the V array used in step L of the computations, and I and J are the indices of the species pair referenced in V. When the subroutine is called with L = 0, the other arguments MV, I, and J are set internally by the subroutine and need not be furnished by the calling routine; however, all arguments are required when L \neq 0.

In use, subroutine BXSECT is first called with L = 0 to begin the search over the V array. The routine then goes through each step L of the cross section computations, starting with L = 1 and checking for steps with KQ = 9 or 12. The value of MV is accumulated at each step L of the computations from the number of V values NV(KQ(L)) required for that step, where NV(KQ) is a fixed value which has been preset in the code. When a step with KQ = 9 or 12 is found, subroutine BXSECT sets I and J to the indices of the species pair referenced in the V array for that step and returns control to the calling program. The latter program may then reset I and J to any desired values and recall subroutine BXSECT again with the current values of L and MV to enter the revised data in the V array and resume the search of the computations again from the point at which it was left off. It should be noted that for each step with KQ = 12, there are two species pairs referenced in the V array, and returned by subroutine BXSECT, while for KQ = 9 there is only one.

When all steps of the computations have been searched, as indicated by the index L of the step becoming equal to the number of steps NKQ in the edited cross section data in common block/TRANS/, the subroutine sets L to 0, as a signal that the search has been completed, and returns control to the calling program.

3.9 Subroutine CØMM

The function of subroutine CØMM is to compute some physical quantities upon which the rates of change of the species concentrations and the other dependent variables depend. The actual calculations of the rates of change are then carried out in subroutine EXACT.

CØMM is called only from subroutine DERIVS, and the call to CØMM is always preceded by a call to subroutine THERM, which computes the species properties at the current nondimensional temperature, $CT = T/T_0$. In the case of an electronic nonequilibrium model, DERIVS calls THERM twice to obtain the required properties at both the gas temperature CT and the electron temperature $CTE = T_e/T_0$.

To facilitate the discussion, the correspondence of some of the Fortran symbols with the mathematical notation used in Volume I is reviewed in the following table. Symbols not listed here are defined in Section 4.77.

ALGJ(J)	$\ln \gamma_j$
BETA(I,J)	β_{ij}
CHI(I)	χ_i
CLKF	$\ln k_f$
CLKR	$\ln k_r$
CLNPI(I)	$\ln P_i$
CLNLMC(I)	$\ln (1 - \chi_i)$
CLNTD(L)	$\ln T$ for $L = 1$; $\ln T_e$ for $L = 2$

CLTBF	$\ln (\rho \sum \gamma_j)$, sum over third bodies
CLZ2	$\ln (R_0 T)$ or $\ln (R_0 T_e)$, argument in cm^3 atm/mole
CLZ3	$\ln (\rho R_0 T)$ or $\ln (\rho R_0 T_e)$, argument in atm-g/mole
CTD(L)	T/T_0 for $L = 1$; T_e/T_0 for $L = 2$
EF	ϵ_{fi}
ER	ϵ_{ri}
GJ(J)	γ_j
PI(I)	P_i
PICHI(I)	$P_i \chi_i$
QDPE	\dot{q}_e
QDPR	\dot{q}_r
QF	q_{fi}
QR	q_{ri}
RHØP	ρ
RHPL	$\ln \rho$
SCPGH	$\frac{1}{R_0} \sum_{j=2}^n \gamma_j c_{pj}$
SU	u/u_s
TEP	T_e
T6	$\sum_{i=1}^n \nu_{ij} \ln \gamma_j$ or $\sum_{j=1}^n \nu'_{ij} \ln \gamma_j$

T $\sum \gamma_j$, sum over third-body species

T8 $\sum_{j=1}^n \nu'_{ij}$

T9 $\sum_{j=1}^n \beta_{ij} \left[\ln (R_0 T') + \frac{\mu_j^0(T')}{R_0 T'} \right]$

where T' is T or T_e

XMJATD(J,L) $\mu_j^0(T')/R_0 T'$, where $T' = T$ for $L = 1$,
 $T' = T_e$ for $L = 2$

XNULIJ(I,J) ν_{ij}

XNULJP(I,J) ν'_{ij}

Z1 $\ln(1/u)$, for u in cm/sec

Subroutine CØMM may be divided, for purposes of explanation, into three parts: an initialization section, a main loop over the reactions, and a final section. These parts of the subroutine will now be discussed.

3.9.1 Initialization Section

This section extends from the subroutine entry point through the statement $RCALC = .FALSE.$ (card CØM 107). It sets up the ALGJ array, computes a number of sums which are required later in CØMM or in other parts of NATA, calculates the nondimensional flow velocity SU from the energy equation I(245), and calls subroutine GEØM to compute the gas density.

In addition, it contains the controls for intermittent dumping of diagnostic data in the nonequilibrium routines. The indicator for these dumps is the integer variable ISW5B in blank common. ISW5B is preset to zero. For the zero

value, no dumps are produced. If ISW5B is set in the input to a positive value, then dumps are produced each time CØMM, EXACT, RNKT, or PRTA is called. These dumps provide a detailed trace of the operation of the nonequilibrium integration, but give several pages of printed output for each integration step. Thus, it is hardly feasible to run a complete solution with continuous dumps. However, operation of the nonequilibrium routines at late stages in the solution can be studied by calling for intermittent dumps. This is done by setting ISW5B to a negative value in the input. Upon the first entry into CØMM, the variable ICYCLE is then reset from its preset value of 90000000 to |ISW5B|. The counter ICØUNT is incremented by 1 every time CØMM is entered. Whenever ICØUNT is an exact multiple of ICYCLE, ISW5B is reset to 1; otherwise, it is reset to 0. Thus, upon every ICYCLEth entry into CØMM, the dumps are switched on and are produced in all of the nonequilibrium routines until CØMM is entered the next time, when they are switched off again. The coding of these controls is in the first nine executable statements of CØMM.

3.9.2 Main Loop Over the Reactions

The main calculations in CØMM are performed in a large DØ loop headed by the statement, DØ 460 I = 1, ISR (card CØM 109). Figure 7 is a flowchart of this main loop. In essence, this loop performs only two functions:

- 1) First, it computes the quantities $P_i \chi_i$ appearing in the rate equations I(287).
- 2) Second, in the case of an electronic nonequilibrium model, it evaluates the energy transfer to the electron gas \dot{q}_e and the radiative loss \dot{q}_r due to the reactions.

The quantities P_i and χ_i are computed from equations I(288) and I(289), respectively, in logarithmic form:

$$\ln P_i = \ln\left(\frac{1}{u}\right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \sum_{j=1}^n \nu_{ij} \ln \gamma_j \quad (39)$$

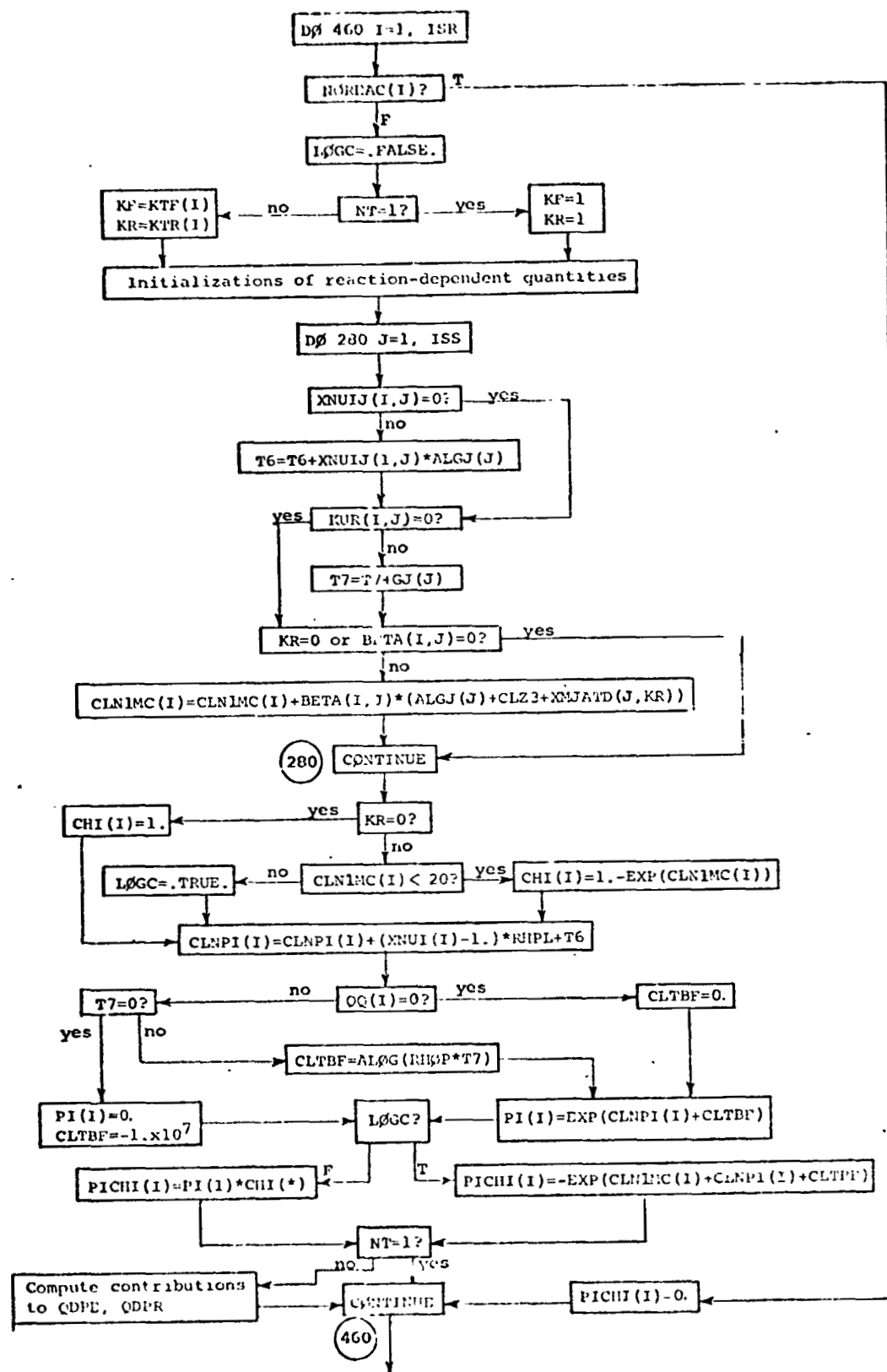


Figure 7. Flow Chart of Main Loop Over Reactions in Subroutine CQMM

ORIGINAL PAGE IS
OF POOR QUALITY

$$\ln (1 - \chi_i) = \beta_i \ln (\rho R_0 T) + \sum_{j=1}^n \beta_{ij} \left[\ln \gamma_j + \frac{\mu_j^0}{R_0 T} \right] \quad (40)$$

In equation (40), the equilibrium constant K_i in I(289) has been expressed in terms of more basic quantities using equation I(278). In the summations indicated in (39) and (40), terms in which the stoichiometric factor ν_{ij} or β_{ij} is zero are omitted.

If $\ln (1 - \chi_i)$ is less than 20, P_i and χ_i are computed from (39) and (40) using

$$P_i = \exp (\ln P_i) \quad (41)$$

$$\chi_i = 1 - \exp [\ln (1 - \chi_i)] \quad (42)$$

and $P_i \chi_i$ is obtained by multiplication. On the other hand, if $\ln (1 - \chi_i)$ is greater than or equal to 20, so that $(1 - \chi_i)$ is greater than 4.85×10^8 , the term 1 in $(1 - \chi_i)$ is neglected and the product $P_i \chi_i$ is computed as

$$P_i \chi_i = - \exp [\ln P_i + \ln (1 - \chi_i)] \quad (43)$$

This algorithm allows the computation of a finite $P_i \chi_i$ value in situations where P_i is so small that an exponential underflow occurs in (41), but χ_i is negative and so large that $P_i \chi_i$ is within the range representable by floating-point numbers. It also makes possible a simple treatment of the special cases which arise when one or more of the γ_j are zero, as discussed below.

If a list of third-body (catalytic) species has been specified for the reaction, the expression I(288) for P_i is replaced by

$$P_i = \left[\frac{\rho^{\nu_i-1}}{u} k_{fi} \prod_{j=1}^n \gamma_j^{\nu_{ij}} \right] \cdot \rho \sum_{tb} \gamma_k \quad (44)$$

in which the sum on the right is over the third-body species specified. The corresponding logarithmic expression is (39) with a term $\ln(\rho \sum_{tb} \gamma_k)$ added to the right. The third bodies do not affect χ_i . Equation (44) may be derived from I(283) and I(287) by noting that, for a catalytic species (k), $\gamma_{ik} = \gamma'_{ik} = 1$. Thus, a factor $\rho \gamma_k$ can be removed from both terms on the right in I(283). If the i th reaction involves more than one third-body species, each such species gives a separate three-body reaction. The effects of these reactions can be summed to give a single reaction with an effective third-body concentration which is the sum of the concentrations of the actual catalytic species, as in (44).

In the case of an electronic nonequilibrium gas model, there are two temperatures, the gas temperature T and the electron temperature T_e . The forward reaction rate k_{fi} can depend on either T or T_e , or both, or neither. The form of temperature dependence of k_{fi} is specified by the array KTF(I). If the indicator $KF = KTF(I)$ is equal to 1, k_{fi} is given by I(69). If KF is equal to 2, the temperature T in I(69) is replaced by T_e . This procedure is coded by placing the two non-dimensional temperatures into a two-element array CTD(L), and placing the logarithms of the two temperatures (expressed in degrees Kelvin) into an array CLNTD(L). Both cases are then treated by a single formula involving CTD(KF) and CLNTD(KF), i.e., statement 190 of CØMM, which evaluates $\ln k_{fi}$ from

$$\ln k_{fi} = \ln C_i + \gamma_i \ln T_k - \frac{E_{ai}}{R_0 T_k} \quad (45)$$

where T_k is equal to T for $k = 1$ and equal to T_e for $k = 2$. In (45), $\ln C_i = \ln A_i - \gamma_i \ln(10^4)$.

Three special forms for the forward reaction rate, used in the electronic nonequilibrium model for argon (Appendix A in Volume II), are indicated by $KTF(I) = 3, 4$, and 5 . For $KTF(I) = 3$, k_f is assumed to be given by

$$k_f = C_i T_e^{\gamma_i} (1 - e^{-E_i/R_0 T}) \quad (46)$$

where E_i is a parameter with dimensions of energy. For $KTF(I) = 4$,

$$k_f = \frac{C_i T_e^{\eta_i}}{\max(1, \tau)} \quad (47)$$

where

$$\tau = b \rho \gamma_a R \quad (48)$$

Here R is the local nozzle radius (or equivalent radius in the case of a channel), and γ_a is the concentration of the product atom in the reaction. For $KTF(I) = 5$,

$$k_f = \frac{C_i}{\sqrt{R}} \quad (49)$$

In NATA, the reverse rate constant k_{ri} is calculated from the detailed-balancing relation I(277), using the equilibrium constant K_i , equation I(278). In electronic nonequilibrium gas models, the coding of CØMM allows for three types of temperature dependence of K_i . If the indicator $KR = KTF(I)$ is zero, the reverse reaction is assumed to occur only at a negligible rate. If KR is 1, K_i is calculated from I(278) using the gas temperature T . If $IR = 2$, the temperature T in I(278) is replaced by T_e . The chemical potentials μ_j^0 in I(278) are temperature dependent. During calculations based on an electronic nonequilibrium model, both $\mu_j^0(T)$ and $\mu_j^0(T_e)$ are computed, and these data are stored in the doubly dimensioned array $XMJAT(J,L)$ with dimensions (20,2). The coding of equation (40) utilizes the array with the index $L = KR$ so that both cases can be treated using a single formula, card CØM 153.

Cases in which one or more of the concentrations $\gamma_j = 0$ require some special consideration because of the appearance of $\ln \gamma_j$ in the formulas used in CØMM. In NATA, the γ_j are represented by the double precision array $GJ(J)$. A zero

value for an element of this array could come about in one of two ways:

- (1) One of the concentrations GJ(J) could underflow as a result of its decrease during the flow solution. However, this is most unlikely to occur in practice because, to underflow, GJ(J) must fall below 10^{-78} (on the IBM 360/75) or below 10^{-308} (on the UNIVAC 1108). Since GJ(J) can decrease by no more than 10 percent in a single integration step because of step size controls, the number of steps required to reach underflow from initial reservoir values of (say) 10^{-1} to 10^{-20} is of the order of 1000 or more, even on the IBM 360. Also, unless the control parameter GAMIN were set to zero, the decrease of GJ(J) would be stopped, long before underflow was reached, by the freezing of minor species (Section 7.5.3 of Volume I).
- (2) A zero GJ(J) value could arise accidentally, at an intermediate point in the Treanor-Runge-Kutta integration step, as a result of a negative $\Delta \gamma_j$ which happened to be exactly equal in magnitude to the γ_j value at the start of the step. This is quite unlikely, but it could happen.

To allow for these rather remote possibilities, the coding of CØMM is designed to cope with zero concentrations if they should occur. If one of the species concentrations, γ_k , is zero, four special cases can arise, depending upon the values of the stoichiometric coefficients ν_{ik} , ν'_{ik} in the reaction I(219):

$$\text{Case 1: } \nu_{ik} = \nu'_{ik} = 0 \quad (50a)$$

$$\text{Case 2: } \nu_{ik} = 0, \quad \nu'_{ik} > 0 \quad (50b)$$

$$\text{Case 3: } \gamma_{ik} > 0, \gamma'_{ik} = 0 \quad (50c)$$

$$\text{Case 4: } \gamma_{ik} > 0, \gamma'_{ik} > 0 \quad (50d)$$

In the Cornell Aeronautical Laboratory program from which NATA was derived, each of these cases was treated separately by special coding. In the present version of the code, all of these cases are treated using the formulas (39) to (44) with the aid of a computational artifice.

Cases involving zero species concentrations γ_j require special consideration because of the appearance of $\ln \gamma_j$ in the formulas (39) and (40). Any attempt of the program to evaluate $\ln 0$ would lead to immediate termination of the run by the computer operating system. In subroutine CØMM, the quantities $\ln \gamma_j$ are stored in an array, ALGJ(J). For species whose concentrations γ_j are zero, ALGJ(J) is set to a large negative number, L, whose magnitude is chosen in accordance with the following criteria:

- (1) L must be sufficiently large in magnitude that, even when combined additively with other terms as indicated in (39) and (40), it is sufficiently negative to produce a guaranteed exponential underflow, i.e.,

$$\exp(L + \text{other terms}) = 0 \quad (\text{exactly}) \quad (51a)$$

This will happen so long as the argument of the exponential function is less than about -88 on the UNIVAC 1108 or less than about -180 on the IBM 360/75.

- (2) L must be sufficiently small in magnitude that the operation

$$L + \text{other terms} - L = \text{other terms} \quad (51b)$$

gives a result for "other terms" containing several significant digits of accuracy. The "other

terms" in equations (39) and (40) are all natural logarithms of the values of physical quantities whose values lie roughly in the range 10^{-20} to 10^{20} ; thus, the logarithms range from about -50 to +50. The second condition can thus be satisfied by taking L to be less than about 10^4 or 10^5 . If L were chosen to be much larger, say 10^{10} , then the first addition in (1b) would give $L + \text{other term} = L$ (exactly) on the UNIVAC 1108, because the change in L due to addition of the "other terms" would be less than the accuracy with which L can be represented as a floating point number (about 8 significant figures).

In subroutine CØMM, L has been chosen to be -2.00, a value which clearly satisfies both conditions.

Now assume that a particular concentration γ_k is zero. With $\ln \gamma_k = L$, the formulas (39) and (40) treat the four cases (50) as follows:

In Case 1, ν_{ik} and β_{ik} are both zero, so that the term $\ln \gamma_k$ does not actually appear in either (39) or (40). In this case, according to (50a), the species k whose concentration γ_k is zero does not participate in the reaction as either a reactant or a product.

In Case 2, $\ln \gamma_k$ does not appear in (39), so that the evaluation of $\ln P_i^k$ proceeds normally. However, $\beta_{ik} = \nu'_{ik}$ is positive, so that the right hand side of (40) contains a term $\nu'_{ik} L$ which is negative and large in magnitude. The resulting exponential underflow during the evaluation of (42) then gives $\chi_i = 1$. This is the correct result for this case, since $\gamma_k^{\beta_{ik}-1}$ is a zero factor in the second term on the right in equation I(289).

In Case 3, equations (39) and (40) can be written

$$\ln P_i = \ln \left(\frac{1}{U} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \sum_{j \neq k} \nu_{ij} \ln \gamma_j + \nu_{ik} L \quad (52a)$$

$$\ln(1 - \chi_i) = \beta_i \ln(\rho R_0 T) + \sum_{j \neq k} \beta_{ij} \left[\ln \gamma_j + \frac{\mu_j^0}{R_0 T} \right] - \nu_{ik} \left(L + \frac{\mu_k^0}{R_0 T} \right) \quad (52b)$$

Here the term $-\nu_{ik} L$ on the right in (52b) is positive and very large; thus, equation (42) is not used and the product $P_i \chi_i$ is evaluated using (43), which gives

$$\begin{aligned} P_i \chi_i &= - \exp \left\{ \ln \left(\frac{1}{u} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \beta_i \ln(\rho R_0 T) \right. \\ &\quad \left. + \sum_{j \neq k} \nu'_{ij} \ln \gamma_j + \sum_{j=1}^n \beta_{ij} \frac{\mu_j^0}{R_0 T} \right\} \\ &= - \frac{k_{fi} \rho^{\beta_i + \nu_i - 1}}{K_i} \prod_{j \neq k} \gamma_j^{\nu'_{ij}} \end{aligned} \quad (53)$$

The terms involving L have cancelled out. The validity of equation (53) for Case 3 can be verified directly by multiplying equation I(288) and I(289), letting $\gamma_k \rightarrow 0$, and taking (50c) into account.

In Case 4, equations (39) and (40) give

$$\begin{aligned} \ln P_i &= \ln \left(\frac{1}{u} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \sum_{j \neq k} \nu_{ij} \ln \gamma_j \\ &\quad + \nu_{ik} L \end{aligned} \quad (54a)$$

$$\begin{aligned} \ln(1 - \chi_i) &= \beta_i \ln(\rho R_0 T) + \sum_{j \neq k} \beta_{ij} \left[\ln \gamma_j + \frac{\mu_j^0}{R_0 T} \right] \\ &\quad + (\nu'_{ik} - \nu_{ik}) \left[L + \frac{\mu_k^0}{R_0 T} \right] \end{aligned} \quad (54b)$$

Addition of these two equations gives

$$\begin{aligned} \ln [P_i (1 - \chi_i)] &= \ln \left(\frac{1}{u} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \beta_i \ln (\rho R_0 T) \\ &+ \sum_{j \neq k} \nu'_{ij} \ln \gamma_j + \sum_{j \neq k} \beta_{ij} \frac{\mu_j^0}{R_0 T} + \beta_{ik} \frac{\mu_k^0}{R_0 T} \\ &+ \nu'_{ik} L \end{aligned} \quad (55)$$

Since $\nu'_{ik} L$ is negative and large in magnitude, (55) implies that

$$P_i (1 - \chi_i) = P_i - P_i \chi_i = 0 \quad (56)$$

Also, (54a) and (41) give $P_i = 0$; thus, $P_i \chi_i = 0$. This result is correct for Case 4, as may be seen by setting one of the γ_k to zero in I(286) and taking (50d) into account.

The preceding analysis can easily be generalized to include cases in which more than one of the concentrations is zero. Thus, the artifice of setting $\ln \gamma_k = L$ when $\gamma_k = 0$ provides a satisfactory treatment of all cases that can arise.

In calculations for a chemical nonequilibrium model ($NT = 1$), the main loop provides only calculations of P_i and $P_i \chi_i$. In calculations for an electronic nonequilibrium model the loop also computes the contributions to the energy transfer to the electron gas (QDPE) and the radiative power loss (QDPR). These computations are provided by the coding below statement 390, and simply consist of evaluation of the formulas

$$(\Delta \dot{q}_e)_i = k_{fi} \epsilon_{fi} \prod_{j=1}^n (\rho \gamma_j)^{\nu_{ij}} - k_{ri} \epsilon_{ri} \prod_{j=1}^n (\rho \gamma_j)^{\nu'_{ij}} \quad (57a)$$

$$(\Delta \dot{q}_r)_i = k_{fi} q_{fi} \prod_{j=1}^n (\rho \gamma_j)^{\nu_{ij}} - k_{ri} q_{ri} \prod_{j=1}^n (\rho \gamma_j)^{\nu'_{ij}} \quad (57b)$$

which are obtained by combining equations I(320) and I(32lb,c). In equations (57), $(\Delta \dot{q}_e)_i$ and $(\Delta \dot{q}_r)_i$ denote the contributions of the i th reaction to \dot{q}_e and \dot{q}_r , respectively. The energy partition parameters ϵ_{fi} , ϵ_{ri} , q_{fi} , q_{ri} are obtained by calling subroutine EPART. Effects of third-body species, if any, are taken into account using the quantity $CLTBF = \ln(\rho \sum_{tb} \gamma_j)$, already evaluated during the computation of $P_i \chi_i$.

3.9.3 Final Section

After the main loop over the reactions has been completed, CØMM checks the value of INEQ. If this indicator is zero (i.e., if the perturbation technique is still being used to generate the flow solution), the RETURN is executed immediately. If INEQ is nonzero (i.e., if the numerical integration of the rate equations has been started), CØMM calculates the non-dimensional pressure, $PRES = p/p_0$. If a chemical nonequilibrium model is being used, the pressure is computed from equation I(273) in the form

$$\frac{p}{p_0} = \frac{\rho}{\rho_0} \frac{T}{T_0} \frac{W_0}{W} \quad (58)$$

If an electronic nonequilibrium model is in use, equation I(323) is used to calculate the pressure in the form

$$\frac{p}{p_0} = \frac{\rho}{\rho_0} \cdot \frac{\gamma_e T_e + \gamma_h T}{T_0} \cdot W_0 \quad (59)$$

In either case, p/p_0 is then divided by $RØBARA = \tilde{\rho}_0 / \rho_0$ to eliminate the effects of gas imperfections, if any, upon the value of the reservoir density, ρ_0 . This is done because gas imperfections are not taken into account in the nonequilibrium integration, as explained in Section 2.5 of Volume I. In addition, the gas entropy is calculated from equation I(251).

If an electronic nonequilibrium model is in use, CØMM computes the contribution to the electron energy transfer \dot{q}_e from elastic collisions, using the formula

$$(\Delta \dot{q}_e)_{\text{elas}} = \sum_{j=2}^n \frac{2W_e}{W_j} \cdot \frac{3}{2} R_0(T - T_e) \frac{n_e \nu_{ej}}{N_0} \quad (60)$$

where for neutral species j , the collision frequency ν_{ej} is given by

$$\nu_{ej} = \frac{4}{3} \sqrt{\frac{8 R_0 T_e}{\pi W_e}} n_j \bar{Q}_{ej}^{-(1,1)} \quad (61)$$

and for ionic species

$$\nu_{ei} = \frac{8}{3} \sqrt{\frac{\pi}{m_e}} \frac{n_i e^4}{(2k T_e)^{3/2}} \ln \left(\frac{k^3 T_e^2}{\pi n_e e^6} \right) \quad (62)$$

These formulas are derived in Appendix A of Volume II; see equations II(25-31). The notation is

W_e, W_j	molecular weights for electron and j^{th} species
$n_e = N_0 \rho \gamma_e$	electron concentration
n_j	concentration of a neutral atom or molecule
n_i	concentration of an ion
$\bar{Q}_{ej}^{(1,1)}$	Maxwell-averaged momentum transfer cross section (assumed the same for all neutrals)
m_e	electron mass
k	Boltzmann constant
N_0	Avogadro number
e	electronic charge (esu)

In NATA, the cross section $\bar{Q}_{ej}^{(1,1)}$ is obtained by calling the function PIØMEG(TEP). The constant factors in (61) and (62) are represented by VIC1, VIC2, and VNC, and are preset in a data statement in CØMM. Equation (60) is evaluated straightforwardly, and the result $QELAS = (\Delta \dot{q}_e)_{elas}$ is added to the \dot{q}_e due to the reactions to obtain the total rate of energy transfer to the electron gas.

3.10 Subroutine CXSECT

Subroutine CXSECT is used by subroutine XSECT to determine the correspondence between species pairs in the master species list for the code and in the rearranged species list stored in the common variable I(J) (see the discussion of subroutine XSECT below). When the subroutine is called with its arguments set equal to the indices of a pair of species in the master species list, it returns the arguments as the indices of the corresponding species pair in the rearranged list. If the original indices are denoted by L,J, the corresponding revised indices are I(L), I(J) if L-J and I(L)-I(J) have the same sign, and are I(J), I(L) otherwise.

3.11 Subroutine DERIVS

During the nonequilibrium solution, the derivatives of the dependent variables, including the species concentrations, are calculated by calling subroutines THERM, CØMM, and EXACT in sequence. If an electronic nonequilibrium gas model is in use, THERM must be called twice, once with $CT = T/T_0$, and once with $CT = T_e/T_0$, to provide species properties at both the gas temperature and the electron temperature. The use of these properties has been explained in the analysis of subroutine CØMM. Also, if the boundary layer is being included in the flow solution, an iteration is necessary to determine the self-consistent solution for the boundary layer displacement thickness δ^* and the derivatives of the flow variables (Section 7.6 of Volume I). These calculations are called for at several locations in subroutines NØNEQ and RNKT. For compactness of coding, these operations are all placed in a separate subroutine, DERIVS.

For an electronic nonequilibrium model ($TN = 2$), DERIVS saves the current value of CT in the location CTSAVE, resets CT to $CTE = T_e/T_0$, and calls THERM to compute the species properties at the electron temperature. The values $\mu_i^0(T_e)/R_0T_e$ are stored directly into XMJATD(I,2). The enthalpy, specific heat, and entropy of the electrons are saved temporarily in the locations HJE, CPJE, and SENTE. Then CT is reset to the value T/T_0 and THERM is called again to compute the species properties at the gas temperature. THERM overwrites the previously calculated data for the electrons (species 1), but these are reset from HJE, CPJE, and SENTE.

For a chemical nonequilibrium model ($NT = 1$), THERM is called only once, to obtain the properties for all the species at the gas temperature, T.

In either case, DERIVS next calls CØMM and EXACT to compute the derivatives of the species concentrations and other dependent variables. It then applies the condition I(407b); if this condition is violated, the indicator FAILED is set to .TRUE. and a RETURN is executed.

The remaining portion of DERIVS controls the iteration to obtain a self-consistent solution for the boundary layer displacement thickness and the derivative $d \ln M/dx$ of the logarithm of the Mach number. If the boundary layer is not being included in the flow solution ($ISW3B = 0$), or if the solution is still in the upstream (uncoupled) region ($IUPD = 1$), this iteration is not performed. The technique used in the iteration is explained in Section 7.6 of Volume I, in the vicinity of equation I(424).

3.12 Subroutine DSMSØL

DSMSØL is the subroutine called for simultaneous solution of systems of linear equations in the calculation of thermochemical equilibrium conditions, in the equilibrium flow solution, and in the nonequilibrium solution. The calling sequence for DSMSØL was embodied in the original program version received from Cornell Aeronautical Laboratory (reference 6), but the subroutine itself was not provided by CAL because it was written in machine language and would not be operable on other types of computer system. To provide the required

capability for solution of systems of equations, subroutine SIMQ from the IBM Scientific Subroutine Package (reference 7) was incorporated into NATA, and a Fortran DSMSØL routine was written to rearrange the data for the equations from the form provided by the CAL routines into the form required by SIMQ.

For a system of linear equations

$$\sum_{k=1}^i a_{jk} x_k = b_j \quad (j = 1, \dots, i) \quad (63)$$

in i unknowns (x_1, \dots, x_i), NATA stores the coefficients a_{jk} and constants b_j in the double precision array AA in unlabelled common. The dimensions of AA are (22,24). The matrix of coefficients a_{jk} is stored as

$$a_{jk} = AA(j,k) \quad \begin{cases} j = 1, \dots, i \\ k = 1, \dots, i \end{cases} \quad (64)$$

The constants b_j are stored as

$$b_j = AA(j, i + 1) \quad (65)$$

Subroutine SIMQ (discussed below) requires that the coefficients a_{jk} be stored columnwise in a singly dimensioned array A, without gaps, and that the constants b_j be stored in an array B. DSMSØL reorganizes the data into this form, using the storage locations of the array AA to contain A and B. The array A is assigned to the first i^2 storage locations in AA, that is, AA(1,1), AA(2,1), ..., AA(22,1), AA(1,2), AA(2,2), ... The array B is assigned the first i storage locations in the 23rd column of AA, that is, AA(1,23), ..., AA(i,23). If the number of equations i is equal to 22, the reorganization leaves the matrix AA unaltered. If i is less than 22, the coefficients a_{jk} are shifted down so as to fill up the first columns of AA without gaps, and the constants b_j are shifted

up into the previously unused 23rd column. After SIMQ has been called, the solution values x_k provided by SIMQ are shifted from the 23rd column into the $(i + 1)$ th column, for use by the calling routine.

The coding of DSMSØL is straightforward, and can be followed easily with the help of the definitions given in the glossary of Fortran symbols (Section 4.80).

3.13 Subroutine DUMP

DUMP is part of the NATA system for dealing with possible errors in code operation. NATA contains numerous validity checks. If one of these is violated, the routine in which the error was detected calls DUMP. This small subroutine sets the indicator ERR to .TRUE. and writes a message giving the name RNAME of the calling routine. It then returns control to the calling routine. The calling routine then executes a RETURN. At each higher-level routine, statements of the form IF(ERR) RETURN pass control up is still higher levels until the main program is reached. In MAIN, when a .TRUE. value of ERR is detected, subroutine DUMPEX is called to print diagnostic data (see below). After the return from DUMPEX, MAIN proceeds as if a normal case completion had occurred; i.e., the job is not terminated unless the input ISW4A is zero.

Originally, the operations now performed by DUMP and DUMPEX were included in a single subroutine (called DUMP). When NATA became so large that it had to be run using overlay on the UNIVAC 1108, the original DUMP routine was divided into two parts, and the larger of these (DUMPEX) was placed into a separate overlay segment so as to minimize the core storage requirement.

3.14 Subroutine DUMPEX

DUMPEX produces the diagnostic dumps when an error has been detected in the operation of the code. The routine contains unlabelled common and many of the labelled common blocks used in NATA. When DUMPEX is executed, most of the variables and arrays listed in these common statements are printed out

in namelist format. In addition, if a binary tape of data for plotting is being written (DATAPE = .TRUE.), DUMPEX backspaces this tape to eliminate all of the records written during the current case.

3.15 Subroutine ELCØND

Subroutine ELCØND computes the electrical conductivity SIGMA of the gas in mhos/cm from equation I(98) using the values of $SSIG = (e/k)^2 / \sqrt{T}$ and $Q(1, I, J) = k^{-1} \sqrt{T} \Delta_{ij}^{(1)}$ computed in subroutine TRANSP. In this computation it is assumed that if electrons are present in the gas, they are the first species. If, on the other hand, electrons are not present (IELEC = 0), the electrical conductivity is set to 0.0 and the computation is bypassed.

3.16 Subroutine ELTIME

The function of subroutine ELTIME is to determine and print out the elapsed execution time since the beginning of the run. When NATA is run on the UNIVAC 1108, ELTIME uses the utility routines .SET and TIME, which are provided by the EXEC II operating system. When the code is used on the IBM 360/75 at Avco, the functions of RESET and TIME are simulated by a small Fortran program (called RESET, Sec. 3.58) which utilizes the system subroutine ACUCPU.

ELTIME contains a logical indicator CALLED, which is preset to .FALSE. in a data statement. When ELTIME is called for the first time in a job, a test on CALLED leads to a call to RESET, which initializes the time counter I for subroutine TIME to 0, and ETØ and ET are both set to zero. Also, CALLED is set to .TRUE., so that this part of the routine is bypassed when ELTIME is called on subsequent occasions in the run.

In such subsequent calls, ELTIME sets ETP = ETØ, and then sets ETØ to ET, where ET is the value left in storage by the previous call (unless this argument has been altered by the calling routine). Then ELTIME calls subroutine TIME(I), which returns the elapsed time I since the call to RESET, measured in milliseconds. ELTIME converts this time to seconds

and stores it in ET. Then, unless the argument IP is zero, ELTIME converts ET into minutes and stores the result in ETM, and also computes the elapsed time DET in seconds since the last previous printed output of the time. A message is then printed giving ETM and DET. If the printed message is skipped because IP = 0, ETØ is reset to ETP to maintain its significance as the elapsed time since the printing of the last time message.

3.17 Subroutine EPART

This subroutine computes the parameters governing the partition of reaction energy between radiative loss and energy transfer to the electron gas for each reaction in a gas model which includes electronic nonequilibrium. The arguments in the calling sequence are defined in Section 4.85. The subroutine treats six cases, or types of reactions, which are distinguished by an index IT. In the following discussion of these cases, $\epsilon_f = EF$ and $-\epsilon_r = -ER$ denote the energies gained by the electron gas in one mole of reactions in the forward and reverse directions, respectively; and $q_f = QF$ and $-q_r = -QR$ denote the corresponding energies lost by radiation.

The six cases are

$$IT = 1 \quad \epsilon_f = -a R_0 T_e$$

$$q_f = \epsilon_0 - \epsilon_f$$

$$\epsilon_r = q_r = 0$$

$$IT = 2 \quad \epsilon_f = -\frac{3}{2} R_0 T_e$$

$$q_f = \epsilon_0 - \epsilon_f$$

$$\epsilon_r = q_r = 0$$

$$IT = 3 \quad \epsilon_f = \epsilon_r = q_f = q_r = 0$$

$$IT = 4 \quad \epsilon_f = \epsilon_r = -\frac{3}{2} R_0 T_e$$

$$q_f = q_r = 0$$

$$IT = 5 \quad \epsilon_f = \epsilon_r = \epsilon_0$$

$$q_f = q_r = 0$$

$$IT = 6 \quad q_f = \epsilon_0$$

$$\epsilon_f = \epsilon_r = q_r = 0$$

In these formulas, $\epsilon_0 = E_0$ denotes the energy available for partitioning between the electrons and the radiative losses, R_0 the universal gas constant (1.9872 cal/mole-deg), and T_e the electron temperature ($^{\circ}\text{K}$).

3.18 Subroutine EQCALC

EQCALC computes the conditions of thermochemical equilibrium at specified temperature T ($^{\circ}\text{K}$) and pressure P (atm), using the method explained in Section 6.1 of Volume I. As an aid to following the coding, the correspondence of Fortran and mathematical symbols is reviewed in the following list:

BET(I)	$\nu_{i-c}^* - 1$	(I = i-c)
CDIJ(I,J)	$\bar{\nu}_{i-c,j}$	(I = i-c)
CGI(I)	w_i	
QM(I)	q_i	

$$XMJAT(I) \quad \mu_i^0/R_0T$$

$$ZCAP(I) \quad X_i$$

$$ZGMU(I) \quad \ln X_i$$

$$ZPZ \quad \ln p$$

EQCALC first saves the current value of the nondimensional temperature CT in the location CTSAVE. This is done because EQCALC may be called with temperatures T differing from the temperature at the current flow point (for example, during the equilibrium normal shock solution). The nondimensional temperature CT is then reset to T/T₀ to allow calculation of the species thermal properties by calling subroutine THERM.

Next, EQCALC computes initial estimates of the species mole fractions and related quantities. The mole fractions for the independent species (I = 1 to ISC) are set to q_i = QM(I), defined by equation I(10). If any of the QM(I) are zero or negative (due to roundoff errors), the corresponding ZCAP(I) values are reset to 10⁻³. Subroutine THERM is then called to compute the thermal properties of all the species at temperature T. The mole fractions for the dependent species (I = ISC + 1 to ISS) are then estimated using equation I(224), which can be written in the logarithmic form

$$\ln X_i = CHII(i-c) + \sum_{j=1}^c \bar{v}_{i-c,j} \ln X_j \quad (66)$$

where

$$CHII(i-c) = -\frac{\mu_i^0}{R_0T} + (\nu_{i-c}^* - 1) \ln p + \sum_{j=1}^c \bar{v}_{i-c,j} \frac{\mu_j^0}{R_0T} \quad (67)$$

In this calculation, SKIL(i) is used as an intermediate variable, first to represent $\ln X_{i-c}$, then X_{i-c} . If the number of dependent species, ISMC = n-c, is zero, then the calculation of

the dependent species mole fractions is skipped here and elsewhere in EQCALC.

Once initial estimates of the X_i and $\ln X_i$ have thus been determined, EQCALC enters the Newton-Raphson iteration based on equations I(277) to I(233). The iterative loop runs from statement 70 down to (but not including) statement 220. The constant terms F_j , equation I(227), in the system of linear equations I(233a) for the corrections h_n^r are loaded into the $M1^{th}$ column of the matrix AA by the DØ loop ending at statement 90. Note that, in statement 80, X_i for $i = c + 1$ to n is denoted by SKIL(i). Then the matrix of coefficients $-X_n^r (\partial F_j / \partial X_n)^r$, equation I(233b), is loaded into the first ISC rows and columns of AA by the nest of DØ loops ending at statement 140. Subroutine DSMSØL is then called to solve the system of equations for the correction factors h_n^r . Upon the return from DSMSØL, the solution values for these quantities are obtained from the $M1^{th}$ column of the matrix AA, where $M1 = ISC + 1$.

The corrected values of the species mole fractions are then calculated. The mole fractions for the independent species ($i = 1$ to c) are obtained using equation I(234):

$$X_k^{r+1} = (1 + h_k^r) \cdot X_k^r \quad (68)$$

The factor $1 + h_k^r$ is denoted by ZA. If ZA is zero or negative, the corrected mole fraction is estimated by halving the previous value:

$$X_k^{r+1} = X_k^r / 2 \quad (69)$$

These calculations are done in the DØ loop ending at statement 180.

The corrected mole fractions for the dependent species are calculated from the equilibrium relations I(224), based on the corrected X_i for the independent species. Equations (66) and (67) are again used, in the DØ loop ending at statement 200.

The convergence test for the Newton-Raphson iteration is that the absolute values of all the h_k^r be less than or equal to a criterion value TEST, which is preset (in subroutine INIT) to 10^{-6} . If convergence has not been achieved after NTEST = 100 iterations, the DUM routine is called to terminate the case and produce diagnostic output.

When convergence has been achieved, EQCALC computes a number of gas properties based on the temperature T, the pressure P, and the equilibrium mole fractions determined by the Newton-Raphson solution of equations I(227). The nondimensional molar entropy of the gas mixture, $ZSEN = S/R_0$, is calculated using equation I(30b). Note that the specific entropy of the gas is equal to

$$s = \left(\frac{S}{R_0}\right) \cdot \frac{R_0}{W} = \frac{ZSEN \cdot CRA}{ZCM} \quad (70)$$

in cal/gm °K. The nondimensional molar enthalpy ZCH of the gas is calculated from equation I(238) and the molecular weight ZCM from equation I(236). Finally, the corrections for gas imperfections, equations I(79) and I(81), are applied.

3.19 Subroutine EXACT

This subroutine computes the derivatives of the species concentrations and the other dependent flow variables for use in the nonequilibrium solution. EXACT consists of two sections. The first section (down to statement 100) calculates the derivatives required in the perturbation solution; the second gives those used in the nonequilibrium integration.

3.19.1 Derivatives for the Perturbation Solution

The derivatives required in the perturbation solution (Section 7.3 of Vol. I) are those of the equilibrium flow variables. These derivatives are computed in EXACT by solving a system of $n+2$ linear equations of the form

$$\sum_{j=1}^{n+2} A_{ij} X_j = B_i \quad (71)$$

The unknowns X_j are

$$X_j = \frac{d \bar{\gamma}_j}{dx} \quad \text{for } j = 1 \text{ to } n \quad (72a)$$

$$X_{n+1} = \frac{1}{T_0} \frac{dT}{dx} \quad (72b)$$

$$X_{n+2} = \frac{d \ln \bar{\rho}}{dx} \quad (72c)$$

The system of equations is

$$i=1 \text{ to } c: \quad \sum_{j=1}^n \alpha_{ji} \frac{d \bar{\gamma}_j}{dx} = 0 \quad (73a)$$

$$\begin{aligned} i=c+1 \text{ to } n: \quad & \sum_{j=1}^c \frac{\bar{\nu}_{i-c,j}}{\bar{\gamma}_i} + \frac{d \bar{\gamma}_j}{dx} - \frac{1}{\bar{\gamma}_i} \frac{d \bar{\gamma}_i}{dx} \\ & + \frac{T_0}{\bar{T}} \left[\frac{\bar{H}_i/R_0 T_0}{\bar{T}/T_0} - 1 - \sum_{j=1}^c \bar{\nu}_{i-c,j} \left(\frac{\bar{H}_j/R_0 T_0}{\bar{T}/T_0} - 1 \right) \right] \frac{d(\bar{T}/T_0)}{dx} \\ & + (\gamma_{i-c}^* - 1) \cdot \frac{d \ln \bar{\rho}}{dx} = 0 \end{aligned} \quad (73b)$$

$$\begin{aligned} i=n+1: \quad & \sum_{j=1}^n \frac{d \bar{\gamma}_j}{dx} + \frac{T_0}{\bar{T}} \left[\sum_{j=1}^n \bar{\gamma}_j \right] \frac{d(\bar{T}/T_0)}{dx} \\ & + \left[\frac{1}{W} - \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{T_0}{\bar{T}} \right] \frac{d \ln \bar{\rho}}{dx} \\ & = \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{T_0}{\bar{T}} \cdot \frac{d \ln A}{dx} \end{aligned} \quad (73c)$$

$$\begin{aligned} i=n+2: \quad & \sum_{j=1}^n \frac{\bar{H}_j}{R_0 T_0} \frac{d \bar{\gamma}_j}{dx} + \left[\sum_{j=1}^n \bar{\gamma}_j \frac{\bar{C}_{pj}}{R_0} \right] \frac{d(\bar{T}/T_0)}{dx} \\ & - \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \frac{d \ln \bar{\rho}}{dx} = \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{d \ln A}{dx} \end{aligned} \quad (73d)$$

Equation (73a) is I(363) with the meanings of the indices interchanged; equation (73b) is I(367) with the signs of all terms reversed and the index replacements $j \rightarrow i$, $l \rightarrow j$; equation (73c) is I(299) applied to the equilibrium flow; and (73d) is I(293) applied to the equilibrium flow. Note that

$$\frac{\bar{H}_i}{R_0 T_0} = SHJ(i) \quad (74a)$$

$$\frac{\bar{T}}{T_0} = CT \quad (74b)$$

$$\left(\frac{\bar{u}}{u_s}\right)^2 = SU2 \quad (74c)$$

$$\frac{u_s^2}{R_0 T_0} = \frac{R_0 T_0}{W_0} \cdot \frac{1}{R_0 T_0} = \frac{1}{W_0} \quad (74d)$$

Equation I(296) is used in the coding of the quantity $1/W$ in (73c).

The coefficients A_{ij} for these equations (73) are loaded into the array $AAA(I,J)$ for $I = 1$ to $n + 2$ and $J = 1$ to $n + 2$. The constants B_i are loaded into $AAA(I, n + 3)$ for $I = 1$ to $n + 2$. The system of equations is then solved by calling subroutine $DLSOL$, and the resulting values for the derivatives (72) are retrieved from the $(n + 3)$ th column of AAA .

3.19.2 Derivatives for the Numerical Integration

During the numerical integration of the rate equations, the derivatives of the species concentrations are given explicitly by equation I(287). These derivatives are computed in the nest of $D\emptyset$ loops ending at statement 120; they are denoted by $DGJ(J)$.

The nature of the remaining derivatives to be determined depends upon the region of solution (upstream or downstream) and the type of gas model being used. The unknowns solved for in the various cases are summarized in Table I. When an electronic nonequilibrium model is being used (NT=2), there is a fourth dependent variable in the integration, the total enthalpy h_0 . However, its derivative is given explicitly by equation I(333), so that it is not involved in the simultaneous solution for the derivatives listed in Table I.

For a given type of gas model, the same set of equations is solved regardless of whether the solution is in the upstream region (inverse method) or the downstream region (direct integration). However, the matrix of coefficients and the vector of constants differ in the two cases because of the difference between the sets of unknowns being solved for. Figure 8 is a flowchart of the solution for the derivatives. By following this flowchart, one can easily write out the equations being solved in each of the four cases listed in Table I, and verify that they are identical with the text equations listed in the table. It should be noted that $NE = NT + 1$ is the number of unknowns and equations in each case. For $NT = 2$, the variable SCPGH contains the sum $\sum_{j=2}^N C_{pj} \gamma_j$, but for $NT = 1$ the sum is over all species (see subroutine CØMM).

3.20 Function EXP

The operating system on the IBM 360/75 at Avco produces several lines of diagnostic output whenever the Fortran function DEXP(X) underflows to give a value 0. To avoid such interruptions of the normal output, the IBM 360 version of NATA contains a Fortran-coded exponential function, EXP(X), which checks the argument X and sets the result to zero for $X \leq -180$. For $X > -180$, EXP calls DEXP to compute EXP. This routine is not used in the UNIVAC version of NATA because the UNIVAC system EXEC II does not produce a diagnostic message for exponential underflow.

TABLE I

DERIVATIVES DETERMINED BY EXACT DURING THE NUMERICAL INTEGRATION

Region	IUPD	Model Type	NT	Unknown Derivatives	Equations Solved
Upstream	1	Chemical	1	$\frac{d(T/T_0)}{dx}, \frac{d \ln \tilde{A}}{dx}$	I(293), I(299)
Upstream	1	Electronic	2	$\frac{d(T/T_0)}{dx}, \frac{d \ln \tilde{A}}{dx}, \frac{d(T_e/T_0)}{dx}$	I(328), I(331), I(332)
Downstream	0	Chemical	1	$\frac{d(T/T_0)}{dx}, \frac{d \ln \rho}{dx}$	I(293), I(299)
Downstream	0	Electronic	2	$\frac{d(T/T_0)}{dx}, \frac{d \ln \rho}{dx}, \frac{d(T_e/T_0)}{dx}$	I(328), I(331), I(332)

```

graph TD
    Start(( )) --> AAA11[AAA(1,1)=SCPGH]
    AAA11 --> AAA23[AAA(2,3)=CCPJ(1)*GJ(1)]
    AAA23 -- yes --> NT22[NT=22]
    NT22 -- no --> AAA12[AAA(1,2)=-Z1]
    AAA12 --> IUPD12[IUPD=12]
    IUPD12 -- yes --> AAA1NE1[AAA(1,NE+1)=Z1*DLQGR-SHJDC]
    IUPD12 -- no --> NT12[NT=12]
    NT12 -- yes --> AAA14[AAA(1,4)=AAA(1,4)-ECQEF*QDPR]
    AAA14 --> AAA22[AAA(2,2)=-Z1  
AAA(2,NE+1)=-SUMTDG-DLQGR*(CT*SUMGH-Z1+(NT-1)*GJ(1)*CTE)  
AAA(3,2)=0]
    NT12 -- no --> AAA34[AAA(3,4)=GJ(1)*CTE*DLQGR-1.5*CTE*DGJ(1)-ECQEF*QDPE]
    AAA34 --> CALL[CALL DMSQL(AAA,NE,22)]
    CALL --> ERR[ERR?]
    ERR -- T --> RETURN[RETURN]
    ERR -- F --> DT[DT=AAA(1,NE+1)]
    DT --> IUPD07[IUPD=07]
    IUPD07 -- yes --> DLQGR[DLQGR=AAA(2,NE+1)]
    IUPD07 -- no --> IUPD17[IUPD=17]
    IUPD17 -- yes --> DLQGA[DLQGA=AAA(2,NE+1)]
    IUPD17 -- no --> NT27[NT=27]
    NT27 -- yes --> DTE[DT=AAA(3,NE+1)]
    DTE --> End(( ))
    
```

Figure 8. Flowchart for the Derivatives of the Flow Variables in the Numerical Integration

3.21 Subroutine FINDX

The purpose of this subroutine is to determine the value of x at which the geometric area ratio has a given value A . An indicator (UPDOWN) specifies whether the upstream or the downstream solution is desired. An entry point FINDXC(A,MBL,X) permits determining the value of x at which a rectangular channel has a half-width A , in centimeters, on its MBLth profile.

The solution is obtained by a Newton-Raphson iterative procedure. The independent variable in the iteration is taken to be $V = |x|$.

If the area ratio is given (normal entry point), then its value is tested before the solution is attempted. If A is less than 0.99999, the case is terminated by calling DUMP, because by its definition the area ratio cannot be less than unity. If A is between 0.99999 and 1.0, the discrepancy is assumed to be due to a numerical (e.g., round-off) error elsewhere in the program; in this case, X is set to 0.0 (the value for $A = 1.$), and the RETURN is executed.

If A is greater than 1.0, V is initialized to 1.0 and the Newton-Raphson procedure is started. The geometric area ratio AR and its derivative $DADX$ at the point $X = V \cdot \text{sign}(\text{UPDOWN})$ are calculated by calling GEOMAR. If AR agrees with A to within one part in 10^5 , the iteration has converged. If not, a new estimate of V is calculated from the Newton-Raphson formula

$$V = V_0 + \frac{A-AR}{|DADX|} \quad (75)$$

However, V is not allowed to increase by more than a factor of 2 in any step; this is to prevent V from becoming very large after a step in which V was very close to the throat (where $DADX = 0.$). A maximum of 50 iterative steps is allowed.

If the channel width is given (entry FINDXC), then V is initialized to 100 cm, a value typical of axial positions where test panels might be installed in a channel. For each trial value of V, the profile half width $AR = Z(MBL)$ and slope $DADX = DZDX(MBL)$ are computed by calling the entry point GMAR3 of GEOMAR. The same Newton-Raphson formulas and convergence test are used as in the area ratio case.

3.22 Subroutine FRØZEQ

FRØZEQ is the routine which controls the computation of the frozen and equilibrium flow solutions. The frozen flow solution is generated when the routine is called through entry FRØZEN. Entry EQUIL(IPASS) gives the equilibrium solution.

3.22. Nondimensionalization of the Flow Variables

Throughout subroutine FRØZEQ, the calculations are carried out with most of the flow variables in nondimensional form. The relation between the Fortran symbols and the dimensional flow variables is summarized in the following formulas:

AFNX	A_e , effective area ratio
CH	$\frac{W_0 h}{R_0 T_0}$, nondimensional specific enthalpy
CHA	$\frac{W_0 h_0}{R_0 T_0}$, nondimensional enthalpy in the reservoir
CT	T/T_0 , nondimensional temperature
CX	x, axial nozzle coordinate (cm)
FLUX	$\frac{\rho u}{\rho_0 u_s}$, nondimensional mass flux

PRES	$\frac{p}{p_0}$, nondimensional pressure
RHØ	$\frac{\rho}{\rho_0}$, nondimensional density
SU	$\frac{u}{u_s}$, nondimensional velocity

In these definitions, the subscript 0 denotes conditions in the reservoir, R_0 is the universal gas constant, and u_s is a velocity defined by

$$u_s = \sqrt{R_0 T_0 / W_0} \quad (76)$$

where W_0 is the molecular weight in the reservoir and R_0 is the universal gas constant in mechanical cgs units (erg/mole °K).

3.22.2 Determination of Frozen Sonic Conditions

When entry FRØZEN is called, subroutine FRØZEQ determines the sonic conditions in the frozen flow before starting the main frozen flow solution.* This calculation is done by the statements from entry FRØZEN down to entry EQUIL. A flowchart of this part of the subroutine is shown in figure 9. The criterion for locating the sonic point is that the mass flux m be a maximum. This condition follows from the continuity equation I(243) and the fact that the flow area A is a minimum at the sonic throat. The temperature at the sonic point is determined as follows: the temperature is initialized to its reservoir value, and is then decremented repeatedly by the amount $DEL T_1 \cdot T_0$, where $DEL T_1$ is normally 0.01. At each of these temperatures, the corresponding mass flux in frozen flow is calculated by calling subroutines THERM and PRØP. This procedure is continued until the mass flux passes its maximum. Then the temperature step is reduced and an iteration is carried out to locate the sonic-point temperature with high accuracy.

*The calculation of sonic conditions for the equilibrium solution is done in a separate subroutine, NRMAX.

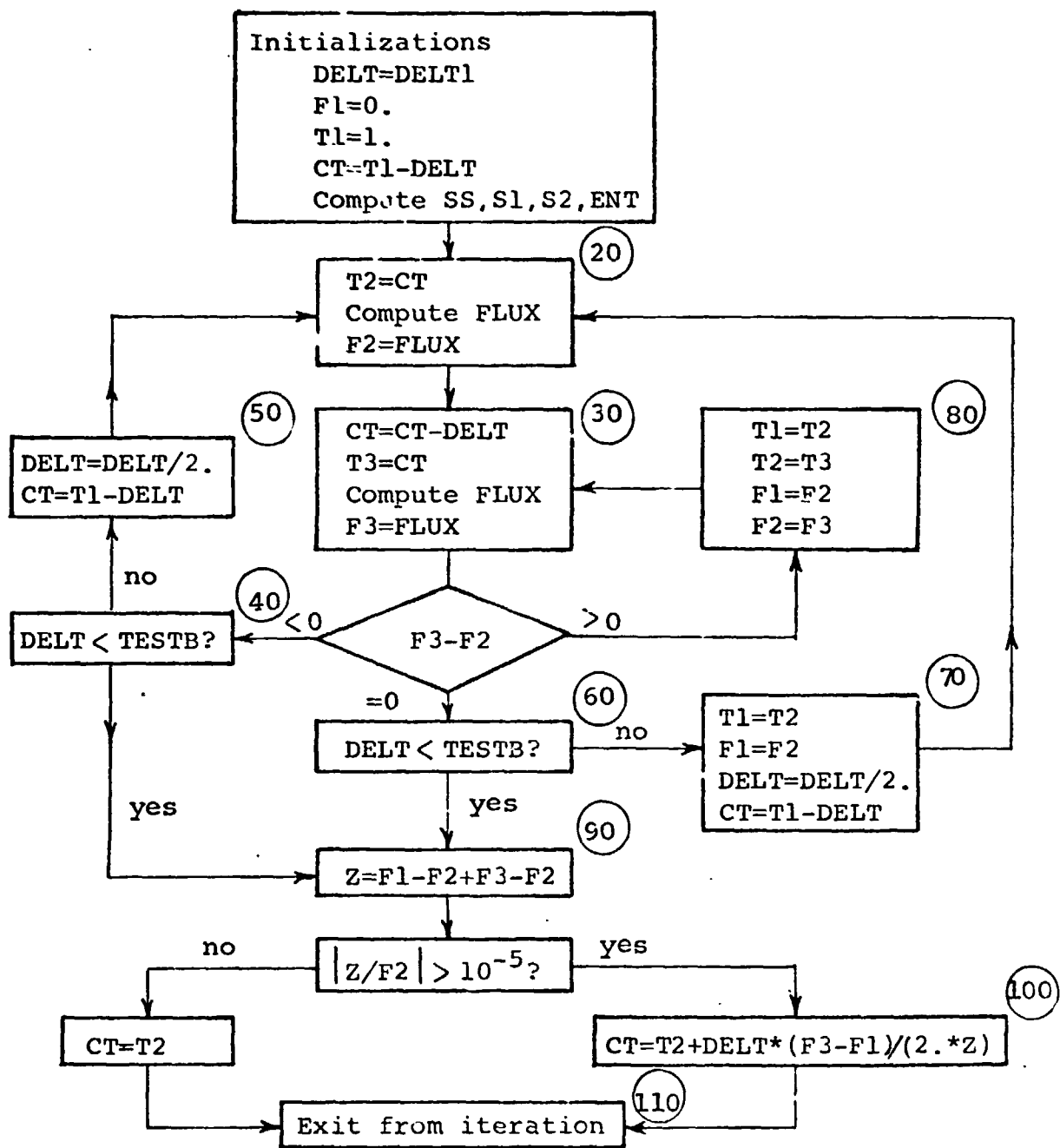
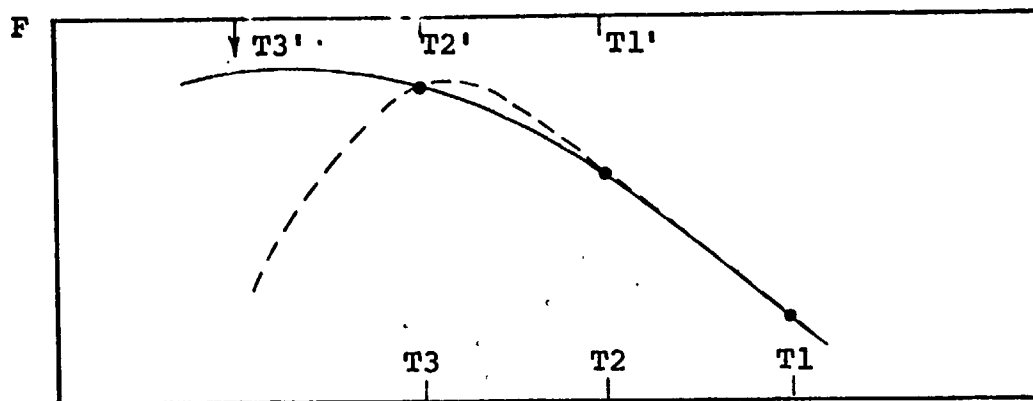


Figure 9. Flow Chart for Determination of Sonic Conditions

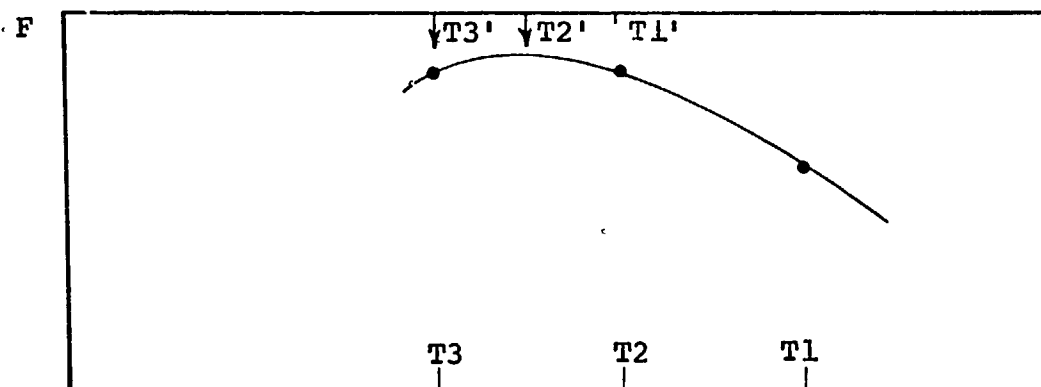
The algorithm for determining the sonic-point temperature (figure 9) is based upon consideration of a sequence of three equally-spaced temperatures $T_1 > T_2 > T_3$ and the corresponding mass flux values F_1, F_2, F_3 . T_1 and T_2 are always chosen in such a way that $F_2 > F_1$; thus, T_1 is definitely known to be higher than the sonic temperature. At each stage of the iteration, the rules for selecting the next triplet of temperatures T_1', T_2', T_3' depend upon the value of $F_3 - F_2$. There are three cases, which are illustrated schematically in figure 10. In case A ($F_3 > F_2$), the three mass flux values form a monotonic sequence $F_3 > F_2 > F_1$. In this case, the available data provide no indication that the point of maximum mass flux has been reached, although this may actually be the case, as shown by the dashed curve in the top section of the figure. In case A, T_1' is set equal to T_2 , T_2' to T_3 , and T_3' is obtained by applying the same decrement $T_1' - T_2'$ to T_2' . In case B ($F_3 = F_2$), the sonic point is known to lie between T_2 and T_3 . In this case, T_1' is set equal to T_2 , the temperature decrement $DELT = \Delta T$ is cut in half, and the remaining two temperatures are set to $T_2' = T_1' - \Delta T$ and $T_3' = T_2' - \Delta T = T_3$. In case C ($F_3 < F_2$), the sonic point could lie between T_1 and T_2 or between T_2 and T_3 . To ensure that T_1' lies above the sonic temperature, it is set equal to T_1 . The decrement ΔT is halved as in case B, and $T_2' = T_1' - \Delta T$, $T_3' = T_2' - \Delta T = T_2$. If the sonic point actually lies between T_2 and T_3 , as illustrated by the solid curve in the bottom part of figure 10, the situation in the next step of the iteration will be as in case A.

At each step, the magnitude of $DELT$ is compared with the criterion value $TESTB = 10^{-5}$. If $DELT < TESTB$, the iteration has converged. A final improvement in the value of the sonic temperature is then obtained by fitting a parabola to the mass flux versus temperature data for the final triplet of points, and determining the maximum analytically based on this quadratic relation. The form assumed is

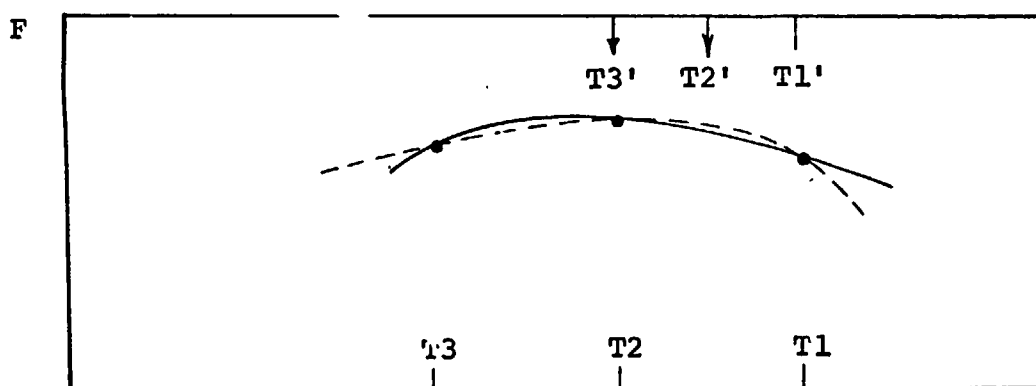
$$F(T) = A + B(T - T_2) + C(T - T_2)^2 \quad (77)$$



Case A: $F_3 > F_2$



Case B: $F_3 = F_2$



Case C: $F_3 < F_2$

Figure 10. Three Cases in the Algorithm for Finding the Throat Temperature

Substitution of the data points (T_1, F_1) , (T_2, F_2) , and (T_3, F_3) and solution for the coefficients A, B, C give

$$F = F_2 - \frac{F_3 - F_1}{2(\Delta T)} (T - T_2) + \frac{F_1 + F_3 - 2F_2}{2(\Delta T)^2} (T - T_2)^2 \quad (78)$$

Setting $dF/dT = 0$ then leads to the final result

$$T_* = T_2 + \frac{(F_3 - F_1)(\Delta T)}{2(F_1 + F_3 - 2F_2)} \quad (79)$$

If the quantity $Z/F_2 = (F_1 + F_3 - 2F_2)/F_2$ is less than or equal to 10^{-5} , equation (77) is not used. In this case, T_* is simply set equal to T_2 , to avoid possible errors due to loss of accuracy on the subtraction in Z (or a divide check in (79) if $Z = 0$).

3.22.3 Main Flow Solution

The basic operation performed by subroutine FRØZEQ is the calculation of the frozen and equilibrium flow solutions. In its essentials, this calculation is quite simple. The nondimensional temperature CT is decremented repeatedly, starting from its reservoir value ($CT = 1.$). At each value of the temperature, the other flow variables are computed, using subroutines THERM and PRØP in the case of frozen flow and subroutine NEWRAP in the case of equilibrium flow. Then the position x in the nozzle at which the flow point occurs is determined from the known nozzle geometry and the value of the area ratio at the flow point, and the results are printed by calling entry ØUT2 of subroutine ØUT1.

The actual complexity of FRØZEQ, as illustrated by its flowchart (figure 11) is due to provisions for treating various options and regions of the solution. The control variables pertaining to some of these cases and conditions will be defined before the subroutine is discussed in detail:

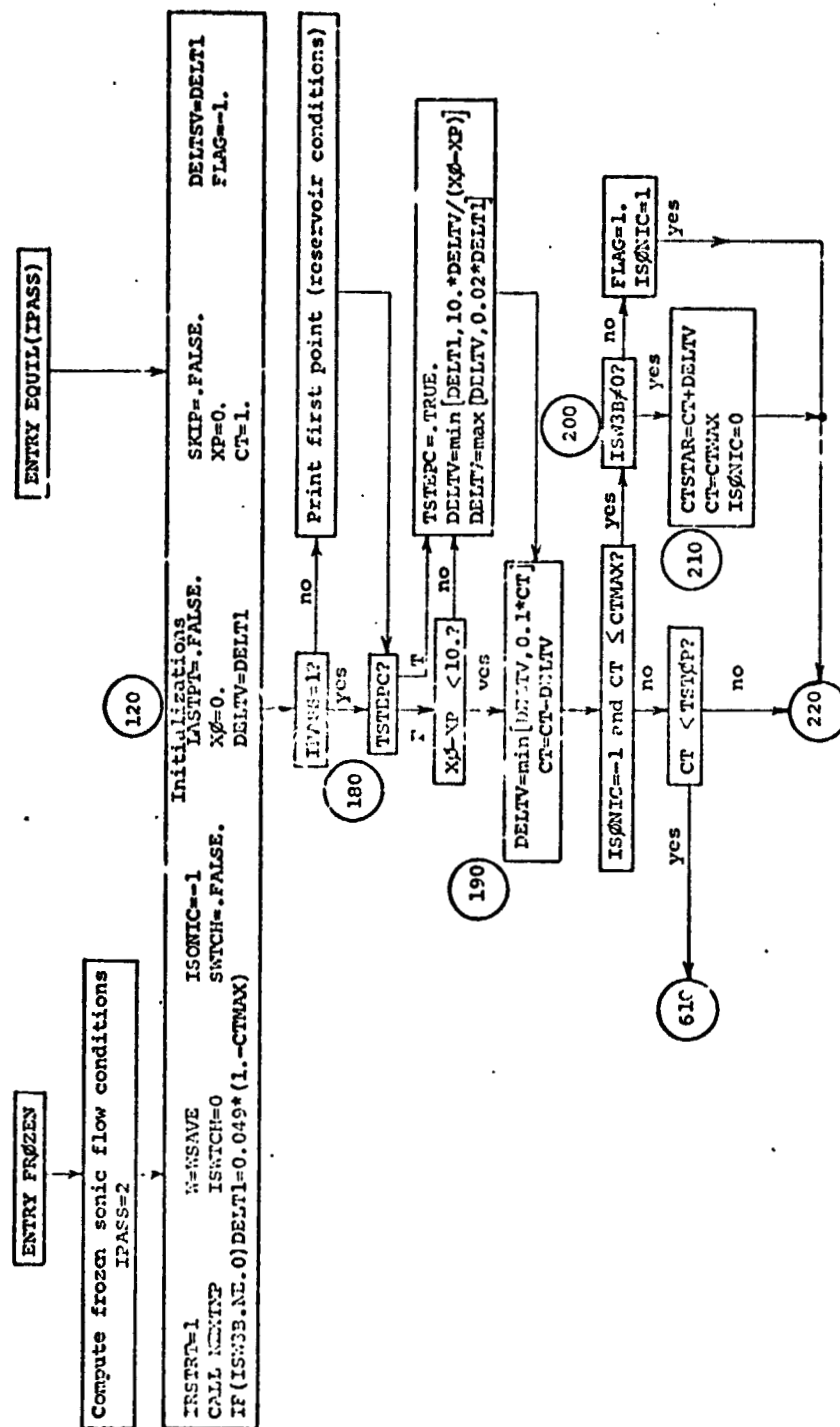
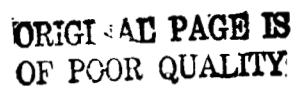


Figure 11a. Flowchart of Subroutine FRQZEQ (Part a)



-14-

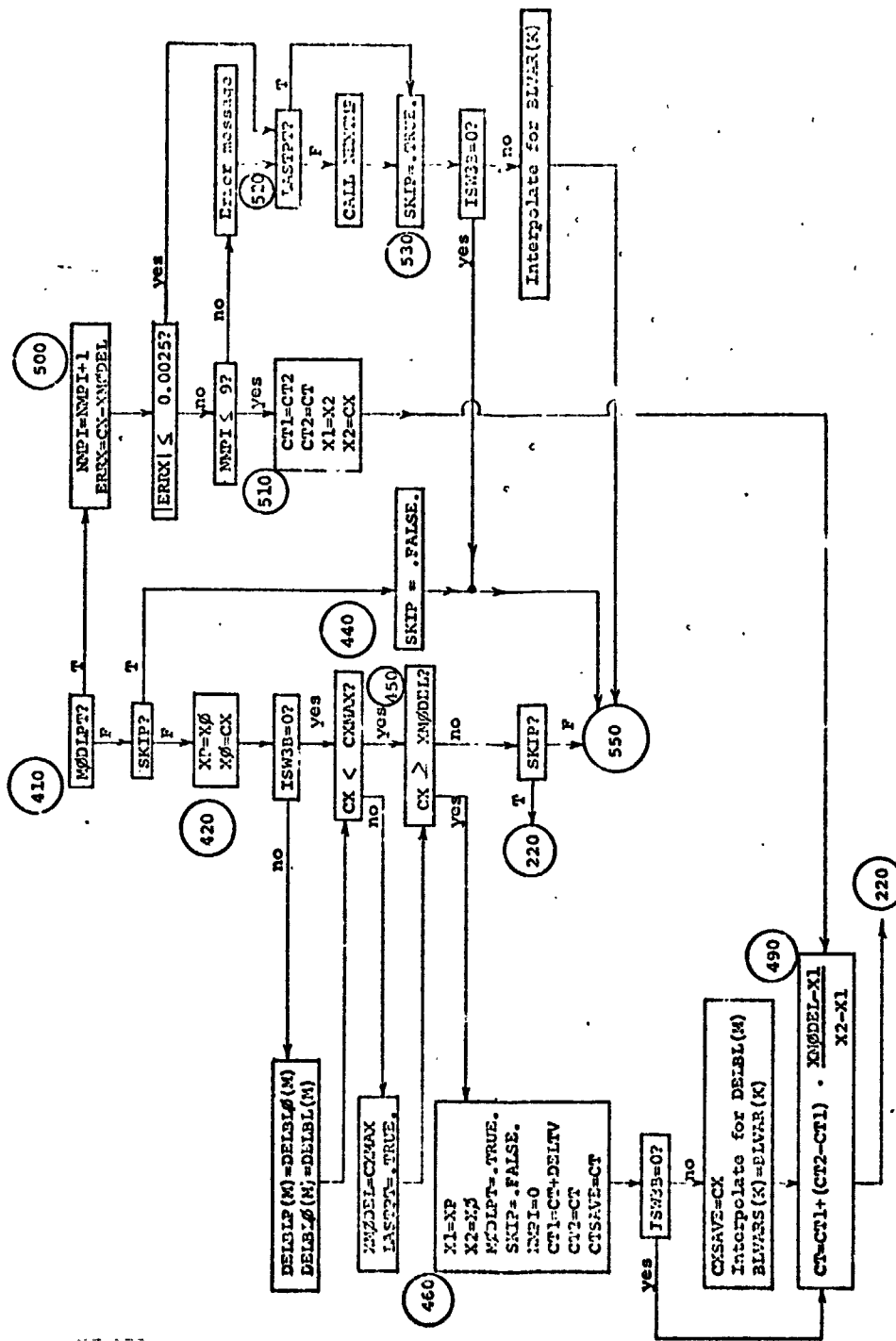


Figure 11c. Flowchart of Subroutine PRQ2BQ (Part c)

ISW3B = 0	Boundary layer is omitted.
ISW3B \neq 0	Boundary layer is included.
IPASS = 1	Preliminary equilibrium solution from the reservoir to the sonic point, to determine the boundary layer displacement thickness at the throat for use in reservoir condition calculations.
IPASS = 2	Final flow solution.
ISØNIC = -1	Region upstream of the throat.
ISØNIC = 0	Sonic point.
ISØNIC = 1	Region downstream of the throat.
MØDLPT = .FALSE.	Regular point of the flow solution.
MØDLPT = .TRUE.	Iteration to find the conditions at a specified model point is in progress.
SKIP = .FALSE.	Regular flow point or model point.
SKIP = .TRUE.	Regular flow point just after a model point.
LASTPT = .FALSE.	Regular flow point or model point.
LASTPT = .TRUE.	Last point of solution.

The simplest case is that of a solution without the boundary layer and without any model points. In this case, FRØZEQ generates a point of the solution by traversing the loop outlined by the statement numbers 180, 190, 220, 310, 410, 420, 440, 450, 550, 580, 600. The temperature decrement DELTV is equal to the input value of DELT1 (normally 0.01) in this case, until the difference $XØ - XP$ in spatial position between two successive flow points exceeds 10 cm. Once this has happened, the unnumbered statements between 180 and 190 reduce DELTV so as to maintain a spatial step size of approximately 10 cm. This variation of DELTV occurs far downstream of the throat, in the region where the free stream temperature has fallen to a small fraction of the reservoir

temperature. DELTV is not allowed to fall below 0.02 DELT1. These adjustments of DELTV in the region far downstream are also made when the boundary layer is included in the solution.

The second statement below 190 tests whether the temperature CT is below the sonic temperature CTMAX, i.e., whether the solution has passed the throat. If it has, in the case without the boundary layer, ISØNIC is reset from -1 to +1, and the indicator FLAG for upstream and downstream solutions in the geometry routines is reset from -1. to +1. Once the flow conditions for the temperature CT have been computed, the spatial position in the nozzle is determined by calling subroutine FINDX(AFN, FLAG, CX) (statement 310). This subroutine determines the upstream (FLAG = -1.) or downstream (FLAG = 1.) position CX at which the geometric area ratio is equal to AFN.

If the boundary layer is included, the input value of DELT1 is saved in the location DELTSV, and DELT1 is reset to $0.049(1. - CTMAX)$ during initialization. This ensures that there will be 20 flow points upstream of the throat. In some cases with $CTMAX \sim 0.9$, the normal input value of $DELT1 = 0.01$ would give only about 10 points above the throat, which would adversely affect the accuracy of the numerical integration in the boundary layer routine BLAYER. Once the throat has been reached, DELT1 is restored to its original value by the statements below 320.

When the boundary layer is included, the effective area ratio $A_e = AFN$ is no longer equal to the geometric area ratio $A_g = AR$, because of the boundary layer displacement thickness, δ^* . The relation between these two area ratios (Section 4.4 of Volume I) involves the displacement thickness at the throat, δ^* . Since this is not known initially, the spatial position CX corresponding to each flow point is calculated on the assumption that $A_g \simeq A_e$ until the throat has been reached. These calculations are done using subroutine FINDX, as in the case without the boundary layer. A special flow point is computed at the throat (ISØNIC = 0, CT = CTSTAR, CX = 0.) in order to determine the quantity

$1 - \delta_*^* = \emptyset MDST$ (statement 390) for the nozzle (or two such values in the case of a channel). Then, beginning at the third point beyond the throat, CX is computed taking the displacement thickness into account, using subroutine AGSØLN. In statement 340, AGSØLN is called with the effective area ratio AFNX for the current step and the displacement thickness DELBL for the preceding step. The resulting CX value is used in the boundary layer calculations (statement below 370). Subroutine AGSØLN is then called again, with the same AFNX and the new DELBL produced by the call to BLAYER. This second call to ACSØLN gives an improved value of CX.

In some problems, at large distances downstream of the throat, the displacement thickness becomes large enough to have a major effect on the effective area ratio. Under such circumstances, the coupling between the boundary layer and the inviscid flow solution can lead to computational instability. The boundary layer routine BLAYER contains a smoothing algorithm intended to suppress such instability (Section 5.11 of Volume I). This algorithm involves a parameter W, preset to a value WSAVE which usually prevents instability. However, occasionally instability develops in spite of the smoothing. In the frozen and equilibrium solutions, the symptom of such instability is that CX begins to decrease with decreasing temperature. When this occurs, subroutine FRØZEQ attempts to produce a valid solution by cutting the stabilizing parameter W in half and restarting the solution in the reservoir. This operation is accomplished by coding between statements 340 and 350. If instability has been encountered after three successive restarts, the solution is abandoned and error dumps are written. Also, restarting is not attempted if the geometric area ratio is more than 4 times greater than the effective area ratio at the point where CX decreases, as in this case the code is probably predicting actual choking of the flow by boundary layer growth.

3.22.4 Model Points

Model points are input-specified spatial locations, in the downstream region of the nozzle, at which special calculations are to be performed. Since model points are

specified by values of CX, while points of the equilibrium or frozen flow solution are specified by values of the temperature, an iteration is required to find the temperature CT corresponding to each model point.

During the initialization, and after the completion of each model point calculation, subroutine NEXTMP is called to determine the location XMØDEL of the next model point. During the main flow solution, after CX has been determined for each flow point, statement 450 tests whether CX is greater than or equal to XMØDEL. If not, the flow solution proceeds normally. If $CX \geq XMØDEL$, the iteration to determine the CT value corresponding to $CX = XMØDEL$ is started. While this iteration is in progress, the indicator MØDLPT is .TRUE..

To avoid complicating the programming of subroutine BLAYER, and to save computer time, BLAYER is not called during the model point iteration. Instead, the boundary layer properties at the model point are approximated by linear interpolation of the data at the preceding and current main flow points, and at each cycle of the iteration subroutine TRANSP is called (statement 360) to determine the transport properties at the current temperature.

The iterative algorithm for improving the estimate of CT corresponding to XMØDEL is a numerical Newton-Raphson technique embodied in statement 490:

$$T = T_1 + (T_2 - T_1) \cdot \frac{X_m - X_1}{X_2 - X_1} \quad (80)$$

where T_2 is the temperature for which calculations were done in the current step, T_1 is the temperature in the preceding iterative step, X_2 and X_1 are the CX values corresponding to T_2 and T_1 , and X_m is XMØDEL.

The convergence criterion for the iteration is that CX differ from XMØDEL by no more than 0.0025 cm or 0.001 inch. A maximum of 10 iterations is allowed. If convergence has not been achieved after 10 iterations, a diagnostic message

is written and the data from the final iteration are used. When the iteration has thus been terminated by either convergence or nonconvergence, the results for the flow conditions at the model point are printed by calling ØUT2, MØDLPT is reset to .FALSE., and subroutine MØDEL is called to perform the required calculations of conditions on models. Also, the indicator SKIP is set to .TRUE., and the next XMØDEL is determined by calling NEXTMP. Before resuming the main flow solution, FRØZEQ checks whether the next XMØDEL is less than or equal to the CX for the current main solution point. If so, the model point iteration is started again to find the flow conditions at the new XMØDEL.

Once all of the XMØDEL values less than or equal to the CX for the current main flow point have been dealt with, the flow conditions for the current flow point are recomputed and printed. The .TRUE. value of the indicator SKIP provides for this recomputation by causing a transfer to statement 220 instead of 550 just below statement 450. Also for SKIP = .TRUE., the determination of CX and the call to BLAYER are bypassed below 220. SKIP is then reset to .FALSE., the results are printed by calling ØUT2, and the main flow solution is resumed.

When a CX greater than or equal to CXMAX is reached, XMØDEL is set equal to CXMAX and the indicator LASTPT is set to .TRUE.. The model point iteration then provides the flow conditions at the position CXMAX. After the iteration has converged and the results have been printed, for LASTPT = .TRUE., control is transferred to statement 610. For nozzle flow problems (NPRFLS = 1), the RETURN is then executed. For channel problems (NPRFLS = 2), subroutine MØDEL is first called with AXISYM = .TRUE. to provide calculation of the stagnation conditions in the channel exit plane. Stagnation condition calculations are suppressed for channels, with this one exception, because stagnation point models and Pitot probes are not used inside channels.

3.23 Subroutine GEØM

The function of GEØM is to calculate the density, the effective area ratio and the derivative of one or the other of these quantities at each position x along the nozzle during

the nonequilibrium flow solution. Two different methods of calculation are used in different parts of the flow solution. During the solution by the perturbation method, and during the direct integration of the rate equations in the region beyond the call to THRØAT, the calculation of the effective area ratio is based directly on the nozzle geometry and the boundary layer displacement thickness, if any, and the density is then calculated from the continuity equation. During the numerical integration by the inverse method upstream of the call to THRØAT, the density is obtained from the analytical density-area relation I(383):

$$\left(\frac{\rho}{\rho_0} A_e\right)^2 \left[1 - \left(\frac{\rho}{\rho_0}\right)^\alpha\right] = C \quad (81)$$

and the effective area ratio is then obtained from the continuity equation I(266). The constants α and C in (81) are determined in the main program from sonic-point conditions, prior to the first call to GEØM, as explained in Section 7.4 of Volume I and in Section 3.1 of the present report.

The overall logical structure of GEØM is diagrammed in figure 12. Immediately upon entry, the subroutine calls GEØMAR to compute the geometric area ratio $S1$ and its derivative $S2$ at the current position CX . If the nonequilibrium flow solution being computed includes the boundary layer ($ISW3B \neq 0$), then AESØLN is called to convert $S1$ into the effective area ratio and $S2$ into the derivative of the effective area ratio, based on the current displacement thickness.

If the flow solution is beyond the call to THRØAT ($IUPD = 0$) or if the perturbation technique is in use ($INEQ = 0$), the effective area ratio $AFNX$ is set equal to $S1$ and the nondimensional density $RHØ$ is calculated from $AFNX$ using the continuity equation. Also, the derivative $DLØGA = d \ln A_e / dx$ is computed from $S1$ and $S2$.

On the other hand, if the numerical integration technique and the inverse method are being used to generate the flow solution ($INEQ = 1$ and $IUPD = 1$), then the nondimensional

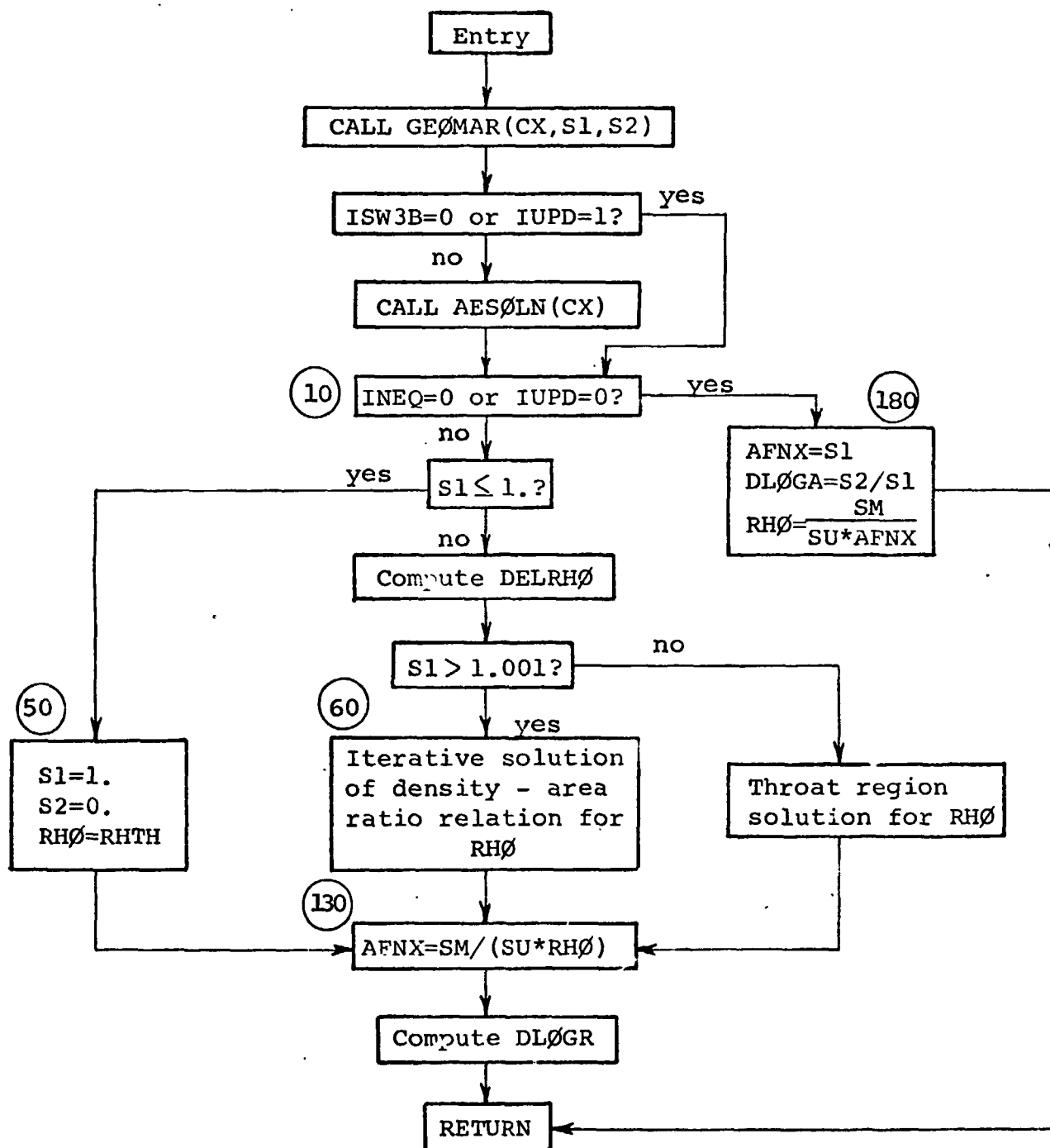


Figure 12. Flowchart of Subroutine GEØM

density $RH\emptyset$ is calculated from $S1$ by solving the density-area ratio relation (81), and the effective area ratio $AFNX$ is recomputed from $RH\emptyset$ using the continuity equation. The derivative $DL\emptyset GR = d \ln(\rho/\rho_0)/dx$ is then computed for use in the rate equations (see the discussion of subroutine EXACT above).

The calculation of $RH\emptyset$ from equation (81) is based on the following analysis. Equation (81) can be solved for A_e :

$$A_e = \frac{\rho_0}{\rho} \sqrt{\frac{C}{1 - (\rho/\rho_0)^\alpha}} \quad (82)$$

The form of this relation is illustrated in figure 13, for the case $\alpha = 0.2$, $C = 0.035049$. The corresponding density ratio at the sonic point is $\rho^*/\rho_0 = 0.62092$. The basic computational problem in $GE\emptyset M$ is to solve (81) or (82) for ρ/ρ_0 , based on a given value of A_e . Except near the throat, the solution is carried out using the Newton-Raphson method; i.e., from a given estimate ρ_p of ρ , an improved estimate ρ_n is obtained as follows:

$$A_p = \frac{\rho_0}{\rho_p} \sqrt{\frac{C}{1 - (\rho_p/\rho_0)^\alpha}} \quad (83a)$$

$$\rho_0 \left(\frac{dA}{d\rho} \right)_p = \frac{\rho_0 A_p}{\rho_p} \cdot \frac{(1 + \frac{1}{2}\alpha)(\rho_p/\rho_0)^\alpha - 1}{1 - (\rho_p/\rho_0)^\alpha} \quad (83b)$$

$$\rho_n = \rho_p + \frac{A_e - A_p}{(dA/d\rho)_p} \quad (83c)$$

Here A_p is the A_e value calculated from (82) for $\rho = \rho_p$; $(dA/d\rho)_p$ is the derivative of (82) evaluated at $\rho = \rho_p$; and equation (83c) is the Newton-Raphson formula.

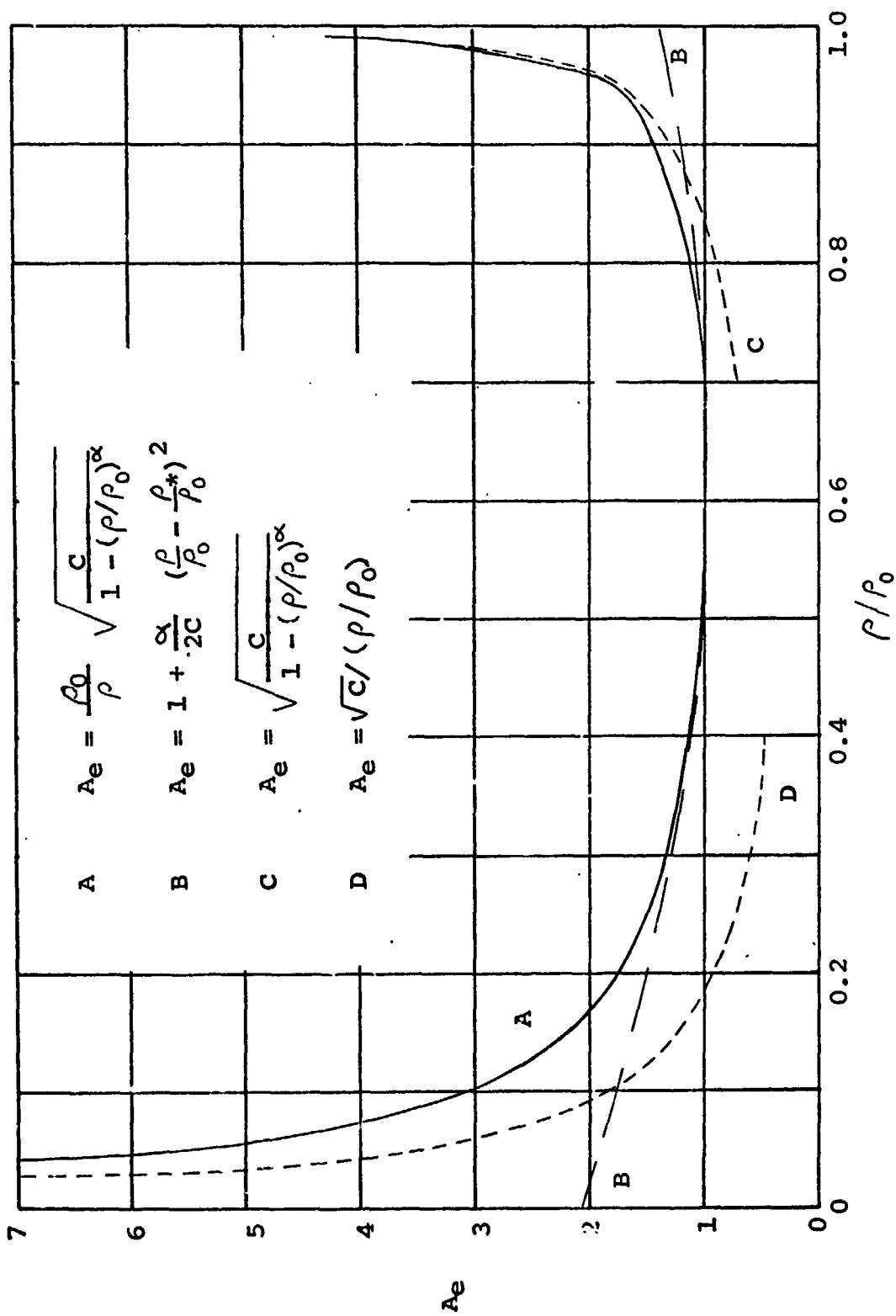


Figure 13. Approximations to the Density-Area Ratio Relation

Examination of equations (83) reveals several potential problems. At the sonic point ($A_e = 1$), the derivative $dA_e/d\rho$ is zero, so that the second term in (83c) is undefined. Near the throat, $(dA/d\rho)_p$ is very small. If a trial value ρ_p is near the throat, and nearer the throat than the solution of (82), the Newton-Raphson formula (83c) can give an "improved" ρ_n which is much less accurate than ρ_p and which may even be outside the permissible range $0 \leq \rho \leq 1$. To deal with these problems reliably, GEOM uses the following algorithm:

- (1) If $A_e \leq 1$, $\rho = \rho_*$ (throat point)
- (2) If $1 < A_e \leq 1.001$ (throat region), ρ is calculated from a Taylor series expansion of (82) around the throat point:

$$\begin{aligned}
 A_e &= 1 + \frac{1}{2} \left(\frac{d^2 A_e}{d\rho^2} \right)_{\rho=\rho_*} (\rho - \rho_*)^2 + \frac{1}{6} \left(\frac{d^3 A_e}{d\rho^3} \right)_{\rho=\rho_*} (\rho - \rho_*)^3 \\
 &= 1 + \frac{\alpha}{2C} \left(\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0} \right)^2 + \frac{(\alpha+1)\alpha^{3/2}}{6C\sqrt{C}(\alpha+2)} \left(\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0} \right)^3
 \end{aligned}
 \tag{84}$$

The coefficients in this equation have been evaluated by differentiation of (82) and use of the sonic-point conditions $\rho = \rho_*$, $A_e = 1$, $(dA_e/d\rho)_* = 0$. In the throat region, where (84) is used, the cubic term is small. If this term were neglected, the solution of (84) would be

$$\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0} = S \sqrt{\frac{2C}{\alpha}} (A_e - 1) \equiv S \cdot (\delta\rho) \tag{85}$$

where S is +1 upstream and -1 downstream of the throat. When the cubic term is taken into account, the solution of (84) can be approximated by

$$\frac{\rho}{\rho_0} - \frac{\rho_*}{\rho_0} = s \left[\frac{2C(A_e - 1)}{\alpha} - \frac{s(\alpha + 1)}{3} \sqrt{\frac{\alpha}{C(\alpha + 2)}} \cdot (\delta\rho)^3 \right]^{1/2} \quad (86)$$

where $(\delta\rho)$ is defined by the second equality in (85). Equation (86) is used to calculate ρ/ρ_0 when $A_e < 1.001$.

- (3) If $A_e > 1.001$ (general case), the Newton-Raphson procedure (83) is used. To ensure its stability, pains are taken to select a first estimate of ρ which is farther from ρ_* than the solution point. Since the relation (82) is everywhere concave upward (figure 13), the Newton-Raphson formula undercorrects ρ whenever the estimate ρ_p lies too far from ρ_* . Thus, if the first estimate ρ_p is too far from ρ_* the iteration converges to the solution without ever overshooting it. Under these circumstances, no instability can develop. To obtain such an initial estimate, the subroutine uses three approximations to (82), all of which lie below (82). One of these approximations is (85). The second is an approximation for $\rho/\rho_0 \sim 1$, namely

$$A_e \approx \sqrt{\frac{C}{1 - (\rho/\rho_0)^\alpha}}$$

from which

$$\frac{\rho}{\rho_0} \approx \left(1 - \frac{C}{A_e^2}\right)^{1/\alpha} \quad (87)$$

The third is an approximation for $\rho \sim 0$,

$$A_e \approx \sqrt{C}/(\rho/\rho_0)$$

from which

$$\frac{\rho}{\rho_0} \approx \sqrt{C}/A_e \quad (88)$$

In the upstream region ($dA_e/dx < 0$), the initial estimate is chosen to be

$$\frac{\rho_p}{\rho_0} = \min \left[\frac{\rho_*}{\rho_0} + \sqrt{\frac{2C(A_e - 1)}{\alpha}}, \left(1 - \frac{C}{A_e^2}\right)^{1/\alpha} \right] \quad (89)$$

In the downstream region ($dA_e/dx > 0$), it is

$$\frac{\rho_p}{\rho_0} = \max \left[\frac{\rho_*}{\rho_0} - \sqrt{\frac{2C(A_e - 1)}{\alpha}}, \frac{\sqrt{C}}{A_e} \right] \quad (90)$$

Once the solution for ρ has been obtained, the derivative $d \ln \rho / dx$ is calculated. Normally this is determined from the formula

$$\frac{d \ln \rho}{dx} = \frac{2C}{\alpha \left(\frac{\rho A_e}{\rho_0} \right)^2 - (\alpha + 2)C} \cdot \frac{d \ln A_e}{dx} \quad (91)$$

which can be obtained by differentiating (82). The factor $d \ln A_e / dx$ on the right is evaluated as $S2/S1$, based on the specified nozzle geometry and displacement thickness. The throat point ($A_e = 1$) is a special case, because there the denominator of (91) is zero and $dA_e/dx = 0$, so that (91) is of the form $0/0$. At the throat, $d \ln \rho / dx$ must be evaluated from (91) using L'Hospital's rule:

$$\left(\frac{d \ln \rho}{dx} \right)_{A_e=1} = 2C \lim_{A_e \rightarrow 1} \frac{d^2 \ln A_e / dx^2}{\frac{2\alpha}{\rho_0} \left(\rho \frac{dA_e}{dx} + A_e \frac{d\rho}{dx} \right)}$$

Since $(dA_e/dx)_* = 0$, this gives

$$\left(\frac{d \ln \rho}{dx} \right)_* = - \sqrt{\frac{C(d^2 \ln A_e / dx^2)_*}{\alpha \rho_* / \rho_0}} \quad (92)$$

The area derivative on the right hand side has to be evaluated from the specified nozzle geometry. In general,

$$\left(\frac{d^2 \ln A_e}{dx^2} \right)_* = \left(\frac{d^2 A_e}{dx^2} \right)_* \quad (93)$$

by virtue of the conditions $A_{e*} = 1$, $(dA_e/dx)_* = 0$. To proceed further, it is necessary to consider the three geometry options separately:

(1) Two-dimensional nozzle

In this case, from equation I(126) with y_* replaced by y_0 as usual,

$$\left(\frac{d^2 A_e}{dx^2}\right)_* = \frac{\frac{1}{y_0} \left(\frac{d^2 y}{dx^2}\right)_* - \frac{1}{y_0} \left(\frac{d^2 \delta^*}{dx^2}\right)_*}{1 - \delta^*/y_0} \quad (94)$$

The term $(d^2 \delta^*/dx^2)_*$ is small, because the displacement thickness δ^* is small and slowly varying at the throat. This term is therefore neglected. Since the throat sections of the profile fit are always of the form I(122) (shape 2, circular arc convex toward the axis), the remaining term in the numerator of (94) is given by

$$\frac{1}{y_0} \left(\frac{d^2 y}{dx^2}\right)_* = \frac{1}{y_0} \left(\frac{d^2 y}{dx^2}\right)_0 = \frac{1}{y_0 P_3}$$

The second equality is obtained by noting that $P_2 = 0$ for a throat section. Thus,

$$\left(\frac{d^2 A_e}{dx^2}\right)_* = \frac{1}{y_0 P_3 (1 - \delta^*/y_0)} \quad (95)$$

(2) Axisymmetric nozzle

From equation I(131) since $(dA_e/dx)_* = 0$ and $A_{e*} = 1$,

$$\begin{aligned} \left(\frac{d^2 A_e}{dx^2}\right)_* = \frac{1}{1 - \delta^*/y_0} & \left[\frac{1}{\sqrt{A_{g*}}} \left(\frac{d^2 A_g}{dx^2}\right)_* \right. \\ & \left. - \frac{1}{2A_{g*}^{3/2}} \left(\frac{dA_g}{dx}\right)_*^2 - \frac{2}{y_0} \left(\frac{d^2 \delta^*}{dx^2}\right)_* \right] \end{aligned} \quad (96)$$

Now dA_g/dx is very small at the sonic point (which is quite near the throat), so that the square of this quantity can be neglected. The quantity $(d^2 \delta^*/dx^2)_*$ is also neglected for reasons discussed above. Hence,

$$\left(\frac{d^2 A_e}{dx^2}\right)_* \simeq \frac{(d^2 A_g/dx^2)_*}{\sqrt{A_{g*}} (1 - \delta^*/y_0)} \quad (97)$$

Evaluation of $(d^2 A_g/dx^2)_*$ using I(116) and I(122) then gives

$$\left(\frac{d^2 A_e}{dx^2}\right)_* \simeq \frac{2}{y_0 p_3 (1 - \delta^*/y_0) \sqrt{A_{g*}}} \quad (98)$$

(3) Rectangular channel

From equation I(134),

$$\begin{aligned} \left(\frac{d^2 A_e}{dx^2}\right)_* = \frac{1}{D} \left[(y - \delta_1^*) \left(\frac{d^2 z}{dx^2} - \frac{d^2 \delta_2^*}{dx^2} \right) + (z - \delta_2^*) \left(\frac{d^2 y}{dx^2} - \frac{d^2 \delta_1^*}{dx^2} \right) \right. \\ \left. + 2 \left(\frac{dy}{dx} - \frac{d\delta_1^*}{dx} \right) \left(\frac{dz}{dx} - \frac{d\delta_2^*}{dx} \right) \right]_* \end{aligned} \quad (99)$$

where

$$D = (y_0 - \delta_{1*}^*) (z_0 - \delta_{2*}^*) \quad (100)$$

At the sonic point, the quantities $\left(\frac{dy}{dx} - \frac{d\delta_1^*}{dx}\right)$ and $\left(\frac{dz}{dx} - \frac{d\delta_2^*}{dx}\right)$ are small, so that their product is considered negligible. The quantities $(d^2 \delta_i^*/dx^2)_*$ are also neglected, as in the other options. Also, $y_* \simeq y_0$ and $z_* \simeq z_0$. Hence,

$$\left(\frac{d^2 A_e}{dx^2}\right)_* \approx \frac{(y_0 - \delta_{1*}^*)(d^2 z/dx^2)_* + (z_0 - \delta_{2*}^*)(d^2 y/dx^2)_*}{(y_0 - \delta_{1*}^*)(z_0 - \delta_{2*}^*)} \quad (101)$$

Evaluation of $(d^2 z/dx^2)_*$ and $(d^2 y/dx^2)_*$ from I(122) gives, finally,

$$\left(\frac{d^2 A}{dx^2}\right)_* \approx \frac{1}{(z_0 - \delta_{2*}^*) P_{3z}} + \frac{1}{(y_0 - \delta_{1*}^*) P_{3y}} \quad (102)$$

where P_{3z} is the P_3 parameter for the z profile and P_{3y} that for the y profile.

3.24 Subroutine GEOMAR

GEOMAR is the basic geometry routine in the NATA code. Through its various entries, it provides calculations of the geometric area ratio and its derivative and of profile ordinates and their derivatives both for nozzles and for rectangular channels.

The subroutine has 4 entries, with the following calls:

GEOMAR(X, ARATIO, DERIVA) (IENTRY = 1)

GMAR(X, Y) (IENTRY = 2)

GMAR2(X, Y, Z) (IENTRY = 3)

GMAR3(X, DYDX, DZDX, Y, Z) (IENTRY = 4)

The arguments have the following meanings:

X = axial coordinate in nozzle (cm)

ARATIO = geometric area ratio

DERIVA = derivative of geometric area ratio (cm^{-1})

Y = profile ordinate (cm)

Z = ordinate for second profile in a channel (cm)

DYDX = derivative of profile ordinate

DZDX = derivative of ordinate for second profile
in a channel

For each profile (I) the subroutine first determines the profile section (J) in which the specified value of X lies. This is the first section for which $X < ATPI(J,I)$. The routine then calculates the profile ordinate from equation I(121), I(122), or I(123) depending on ISHAPE(J,I), and it required also calculates the derivative of the profile ordinate using the derivatives of these formulas:

(1) Straight Line, ISHAPE(J,I) = 1

$$\frac{dy}{dx} = P_2 \quad (103)$$

(2) Circular Arc Convex Downward, ISHAPE(J,I) = 2

$$\frac{dy}{dx} = \frac{x - P_2}{\sqrt{P_3^2 - (x - P_2)^2}} \quad (104)$$

(3) Circular Arc Concave Downward, ISHAPE(J,I) = 3

$$\frac{dy}{dx} = - \frac{x - P_2}{\sqrt{P_3^2 - (x - P_2)^2}} \quad (105)$$

The subroutine then calculates the geometric area ratio, if required, from equation I(118) for a two-dimensional nozzle (NPRFLS = 1, JDIM = 0), I(116) for an axisymmetric nozzle (NPRFLS = 1, JDIM = 1), or I(120) for a rectangular channel (NPRFLS = 2). It calculates the derivative of the area ratio from the derivatives of these formulas, i.e.,

$$\frac{dA_g}{dx} = \frac{1}{y_0} \frac{dy}{dx} \quad (2D \text{ nozzle}) \quad (106)$$

$$\frac{dA_g}{dx} = \frac{2y}{y_0^2} \frac{dy}{dx} \quad (\text{axisymmetric nozzle}) \quad (107)$$

$$\frac{dA_g}{dx} = \frac{y \frac{dz}{dx} + z \frac{dy}{dx}}{y_0 z_0} \quad (\text{channel}) \quad (108)$$

Before executing the RETURN, the routine rescales all of the computed results using the rescaling factor RSA. The area ratio and its derivative are multiplied by RSA in the case of a 2D nozzle, but in an axisymmetric nozzle or channel they are multiplied by $\sqrt{\text{RSA}}$. The factor RSA is initialized to 1.0 and is never changed in equilibrium and frozen flow solutions. In nonequilibrium flow solutions, RSA is reset in subroutine THRØAT so as to make the effective area ratio continuous between the upstream solution by the inverse method and the downstream direct solution.

3.25 Subroutine INGAS

This subroutine has two entry points, INGAS and INTA. The first part of the routine, which is entered by calling INGAS, sets up several arrays which are used in the chemical description of the gas mixture, as discussed in Section 2.1 of Volume I. The mathematical and Fortran notations for these arrays correspond as follows:

A_{ki}	AIN(K,I)	
$\bar{\nu}_{i-c,j}$	CDIJ(I,J)	(I = i - c)
$-1 + \nu_{i-c}^*$	BET(I)	(I = i - c)
q_i	QM(I)	
w_i	CGI(I)	

In the subroutine, the array AIN is first initialized to the square submatrix $\bar{\alpha}_{ij}$ of the matrix α_{ij} with $i = 1$ to c , where α_{ij} is the number of atoms of the j th element

per molecule of the i^{th} species, and c denotes the number of elements in the system. The inverse matrix A_{ki} of $\bar{\alpha}_{ij}$ is then computed and stored in AIN by calling subroutine MATINV.

Next, the matrix $\bar{\nu}$, which gives the composition of the dependent species in terms of the components, is then computed by evaluating equation I(6), and is stored in CDIJ. Then the singly subscripted array $-1 + \nu_{i-c}^*$, where ν^* is defined by equation I(15), is computed in BET. The array q_i giving the invariant composition of the gas in terms of the independent species is calculated using equations I(9) and I(10); it is denoted by QM(I). Finally, the species molecular weights $W_i = \text{CGI}(I)$ are computed as follows:

$$C_i = \sum_{j=1}^c \alpha_{ij} A_j \quad (109)$$

where A_j is the atomic weight of the j^{th} element;

$$W_i = C_i \quad (i = 1, \dots, c) \quad (110a)$$

$$W_i = \sum_{k=1}^c \bar{\nu}_{i-c,k} W_k \quad (i = c + 1, \dots, n) \quad (110b)$$

For the rationale of equation (110b), see equation I(5) and its associated discussion. The molecular weights of the dependent species $i = c + 1, \dots, n$ could also be calculated directly using equation (109), in place of (110b).

The purpose of the calculations following entry INTA is to compute the equilibrium conditions in the upstream reservoir. The actual thermochemical equilibrium calculation is performed by subroutine EQCALC(CTAP,PRESA). The argument CTAP is the reservoir temperature in degrees Kelvin, and PRESA is the reservoir pressure in atmospheres.

The results of the calculations are communicated from EQCALC to INGAS through the common blocks /EQC/ and /EQC2/.

3.26 Subroutine INIT

Subroutine INIT initializes a number of control parameters and nondimensionalizes the species thermochemical data using the reservoir temperature. Since the coding is straightforward, it will not be discussed.

3.27 Subroutine KANDMU

Subroutine KANDMU calculates quantities proportional to the viscosity and the translational thermal conductivity of the gas mixture from equations I(85) to I(87). The first section of the subroutine, down through statement 20, computes the quantities $(25/8)\sqrt{T} \bar{a}_i^{(K)}$ and $(25/8)\sqrt{T} \bar{a}_i^{(\mu)}$ defined by equation I(87), for all species i in the mixture. The value of $(25/8)\sqrt{T} \bar{a}_i^{(\mu)}$ is obtained simply by dividing the factor $ZM2(I)$ out of the quantity $B(I,2)$ computed previously in subroutine KINT, equation (113) below, and is stored again in the array $B(I,2)$, while the quantity $(25/8)\sqrt{T} \bar{a}_i^{(K)}$ is computed directly from I(87) using the matrix elements $Q(3,I,J)$ calculated previously in subroutine TRANSP. The value of $(25/8)\sqrt{T} \bar{a}_i^{(K)}$ is stored in the location $B(I,1)$, replacing the previous quantity set in KINT, which is no longer needed. The reciprocals of the elements $B(I,L)$ are also computed and stored in $BR(I,L)$.

The next section of subroutine KANDMU, from statement 20 down through statement 70, computes the quantities

$$A(1) = (25/8)\sqrt{T} \bar{a}^{(K)}$$

and

$$A(2) = (25/8)\sqrt{T} \bar{a}^{(\mu)}$$

(111)

from equation I(86), using the values of the matrix elements $a_{ij}^{(K)}$ and $a_{ij}^{(\mu)}$ which were previously stored in the lower halves ($I > J$) of the matrices $Q(1,I,J)$ and $Q(2,I,J)$

by subroutine TRANSP. Since the values of $\bar{a}^{(\alpha)}$ for the viscosity and thermal conductivity are both computed from the same equation, the same coding can be used here for both properties, with the index $L = 1$ indicating values computed for the thermal conductivity and $L = 2$ values computed for the viscosity. In connection with this computation, it may be noted that the sum in equation I(86) need only be carried over values of $I > J$, since the matrix $a_{ij}^{(\alpha)}$ is symmetric and the diagonal terms do not contribute to the sum.

The final loop in subroutine KANDMU down through statement 90 computes the values of the subroutine arguments $ZK(1)$ and $ZK(2)$, which are proportional respectively to the translational thermal conductivity and the viscosity of the gas mixture. These values are computed from equation I(85), using the values of the elements $B(I,L)$ and $A(L)$ which were computed earlier in the subroutine

3.28 Subroutine KINT

The statements down through statement 10 of subroutine KINT compute the quantities

$$B(I,1) \equiv \frac{1}{k} \sqrt{T} \sum_{j=1}^n x_j \Delta_{ij}^{(1)} \quad (112)$$

and

$$\begin{aligned} B(I,2) &\equiv \frac{5}{6k} \sqrt{T} \sum_{j=1}^n x_j \Delta_{ij}^{(2)} \\ &= \frac{25}{8} \sqrt{T} ZM2(I) \sum_{j=1}^n x_j A_{ij}^{(\mu)} \end{aligned} \quad (113)$$

which are required in the transport property calculations. Because of the way in which the data $\Delta_{ij}^{(k)}$ are stored in the Q array by subroutine TRANSP, it is necessary to treat the terms with $i < j$ and with $i > j$ separately in these

computations; thus the statement preceding 10 carries out the sum over terms with $i > j$ and statement 10 the sum over terms with $i < j$. The diagonal terms with $i = j$ were added previously during the initialization of the sum by subroutine TRANSP.

Following the computation of the $B(I,1)$ and $B(I,2)$, subroutine KINT uses the computed values of the $B(I,1)$ in the small loop ending at statement 30 to calculate the value of the argument $ZKINT \equiv K_{int}/\sqrt{T}$ in units of milliwatts/cm-°K^{3/2}, where K_{int} is the internal component of the thermal conductivity of the gas given by equation I(95).

3.29 Subroutine LIST

Subroutine LIST prints out certain portions of the input data for each case so as to provide a description of the problem which is to be solved by the code. Subroutine READ also provides some output of similar character.

Portions of the output normally produced by LIST are omitted if they would simply repeat the corresponding output for a previous case. If $SUPGØ = .TRUE.$, the gas model is the same as in the preceding case, and the description of the gas model is skipped. If $SUPGØ$ is $.FALSE.$, the following tables are printed:

- (1) The elemental composition of the gas in terms of "atom fractions", defined as the number of gram-atoms of each element per mole of the mixture of cold species. This table also lists the atomic weights assumed for the elements.
- (2) The data defining the forward rate constant for each of the reactions. This table also includes the criterion value CCHI for switching from the perturbation technique to numerical integration, and the third body matrix for the reactions involving third bodies.

- (3) A table which defines the chemical formula for each species by giving the $\alpha_{ij} = \text{LPIJ}(I,J)$ matrix. A separate column of this table contains 1 if a thermo fit is available for the species, 0 if not.
- (4) The matrix ν'_{ij} of the stoichiometric coefficients for the product side of all the reactions.
- (5) The matrix ν_{ij} of the stoichiometric coefficients for the reactant side of all the reactions.
- (6) A table listing the data for calculations of species thermal properties based on the physical model.
- (7) A table listing the electronic energy levels and degeneracies for all the species for which physical model data are provided.
- (8) A table of the thermo-fit coefficients for those species for which thermo fits are provided.

If an electronic nonequilibrium gas model is being used (NT=2), then a table giving the extra reaction data required by this model is inserted before the thermo-fit table (8).

After the gas model description has been printed out, LIST prints a summary of the input data for model and wedge calculations.

3.30 Subroutine MATINV

This is a matrix inversion subroutine. It is called by INGAS to compute the inverse A_{ki} of the matrix $\bar{\alpha}_{ij}$; see equation I(3) and associated discussion. MATINV computes the inverse of matrix A using the Gauss-Jordan reduction with the maximum pivot strategy (ref. 8).

3.31 Subroutine MØDEL

Subroutine MØDEL is the principal routine for calculations of conditions on models in the NATA code. MØDEL performs normal-shock calculations, computes conditions at the

inviscid stagnation point, calculates stagnation point heat fluxes, and provides output of these results. In addition, if conditions on wedge models are to be calculated, MØDEL calls subroutine WEDGE. The overall structure of MØDEL is diagrammed in figure 14.

3.31.1 General Description

Immediately upon entry, MØDEL performs some checks to determine whether model condition calculations should be done. Such calculations are skipped if both of the logical variables AXISYM and WEDGEM are false, or if the free-stream Mach number is less than 1.5. The latter restriction is imposed because the normal shock solution fails to converge when the Mach number is below some value in the range from 1.5 to about 2. Models are not normally tested at such low Mach numbers in arc-heated wind tunnels because the corresponding nozzle diameter is too small to accommodate models of reasonable size.

If model calculations are to be done, subroutine ELTIME is called with the argument IP = 0 to record the execution time at the beginning of the model calculations. ELTIME is called again, just prior to the RETURN, to determine the CPU time elapsed during the model calculations. This time is printed out. It is usually about 1 to 3 seconds.

Before beginning the normal shock solution, MØDEL sets a number of flow quantities which depend upon the free-stream flow conditions and are thus constant for a given call to MØDEL. When MØDEL is first called in a given flow solution, the equilibrium and frozen shock density ratios EPSLØN and EPSF have the value zero, set in the calling routine (FRØZEQ or NØNEQ). These ratios are reset to the value 0.01. When MØDEL is called with EPSLØN = 0, it computes the Prandtl number at the model surface temperature, PRW, based on the cold species mole fractions QPJ(K). This value of PRW thus assumes that the gas at the model surface is in a state of thermochemical equilibrium at the surface temperature.

Most of the calculations in MØDEL are contained within a large DØ loop with the DØ index ISØLN, beginning below

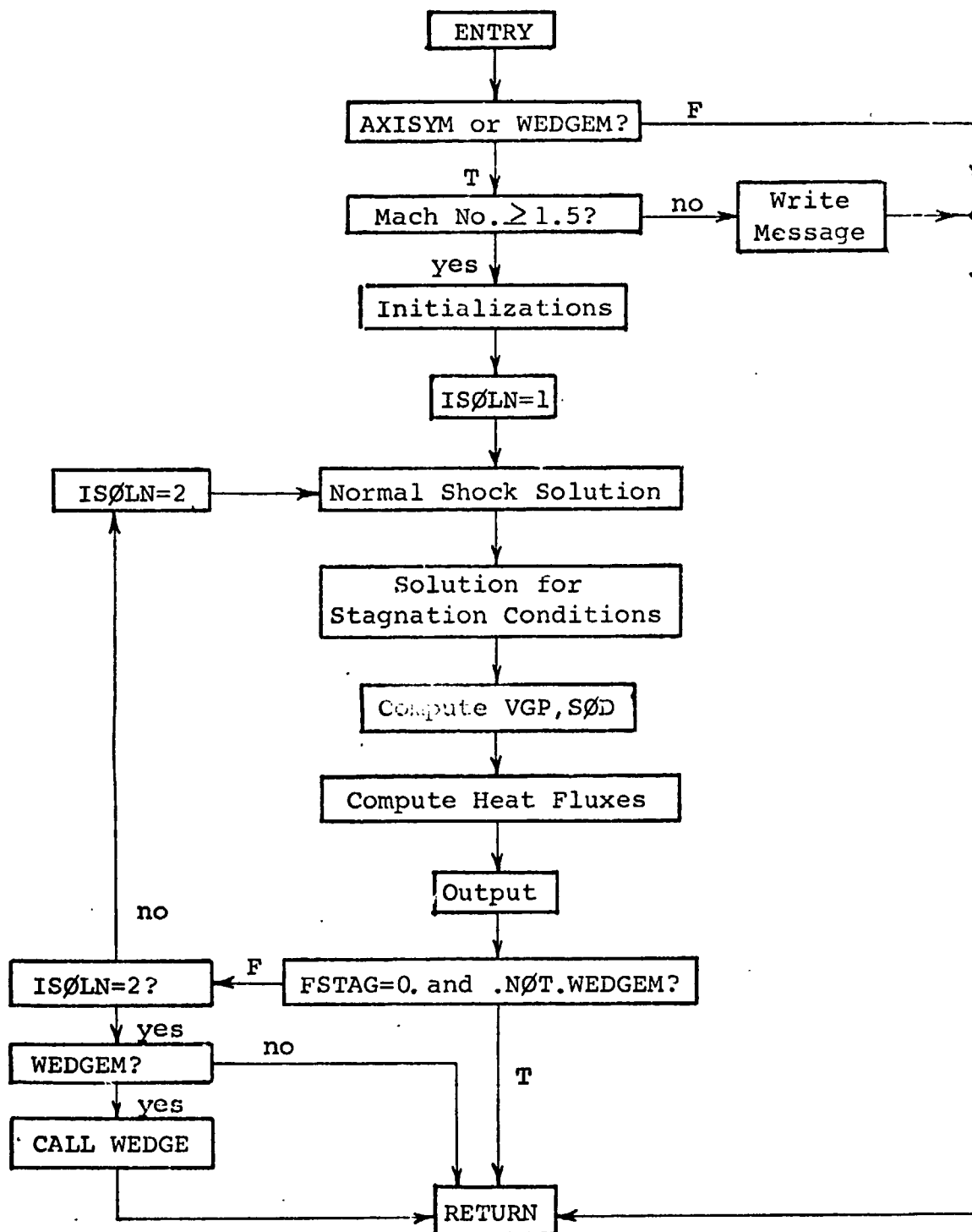


Figure 14. Overall Flowchart of Subroutine MØDEL

statement 40 and ending at 440. For $IS\emptyset LN = 1$, the normal shock is calculated assuming chemical equilibrium behind the shock. For $IS\emptyset LN = 2$, the flow across the shock is assumed to be chemically frozen. If the control variable FSTAG is 0, the frozen shock calculations are skipped unless wedge calculations are to be done ($WEDGEM = .TRUE.$). If wedge calculations are called for, the frozen shock solution is carried out, regardless of the value of FSTAG, to provide the frozen stagnation temperature which is required for use in subroutine WEDGE. If FSTAG is negative, the equilibrium shock calculations are skipped.

For either value of $IS\emptyset LN$, the conditions behind the normal shock and those at the inviscid stagnation point are calculated using methods discussed below in Sections 3.31.2 and 3.31.3. Then the stagnation-point velocity gradient parameter $(R_p/u_1)(du_e/dx)_s$, denoted by $VGP(L)$, is computed for spherical or cylindrical nosed models ($L = 1$) and for flat-faced models ($L = 2$). The ratio $SOD(L)$ of the shock standoff distance to the model radius is also calculated for $L = 1$ and 2. These calculations are based upon equations I(463) to I(465) for the velocity gradient parameter, and I(475) to I(478) for the standoff distance.

Next, the various heat flux values discussed in Section 8.1.4 of Volume I are computed. In preparation for these calculations, the dissociation enthalpy h_p of the gas at the inviscid stagnation point (denoted by HCF) is computed by summing the product of the mole fraction and the enthalpy of formation $SHJA(I)$ over all neutral atomic species (see equation I(544)). In the case of an equilibrium shock, the mole fractions $SAVEC(I)$ at the stagnation point are obtained from the common array $ZCAP(I)$, which is set in subroutine EQCALC. In the case of a frozen shock, the mole fractions are the same as those in the free stream flow ahead of the shock, and are calculated from the concentrations $GJ(I)$ using equation I(1). In the evaluation of HCF, the sum is restricted to neutral atoms by rejecting all species containing more than one element, more than one atom of an element, or a negative number of atoms of an element.* This algorithm

*Positive ions are represented as compounds containing a negative number of electrons.

accepts the electron species, which of course is not a neutral atom, but this does not affect HCF because the formation enthalpy of the electron is zero.

The transport properties μ_e (VISC), N_{Pr} (PR), N_{Le} (LE), and σ (SIGMA) at the inviscid stagnation point are computed by calling subroutine TRANSP. The heat fluxes are calculated using equations I(466), I(468) and I(469) with suitable conversion factors to obtain English engineering units in the output. For example, the numerical coefficient in the Fortran formula corresponding to equation I(466) is obtained as follows:

$$0.51 = \frac{1.8 \times (30.48)^{3/2}}{453.6} \times 0.763$$

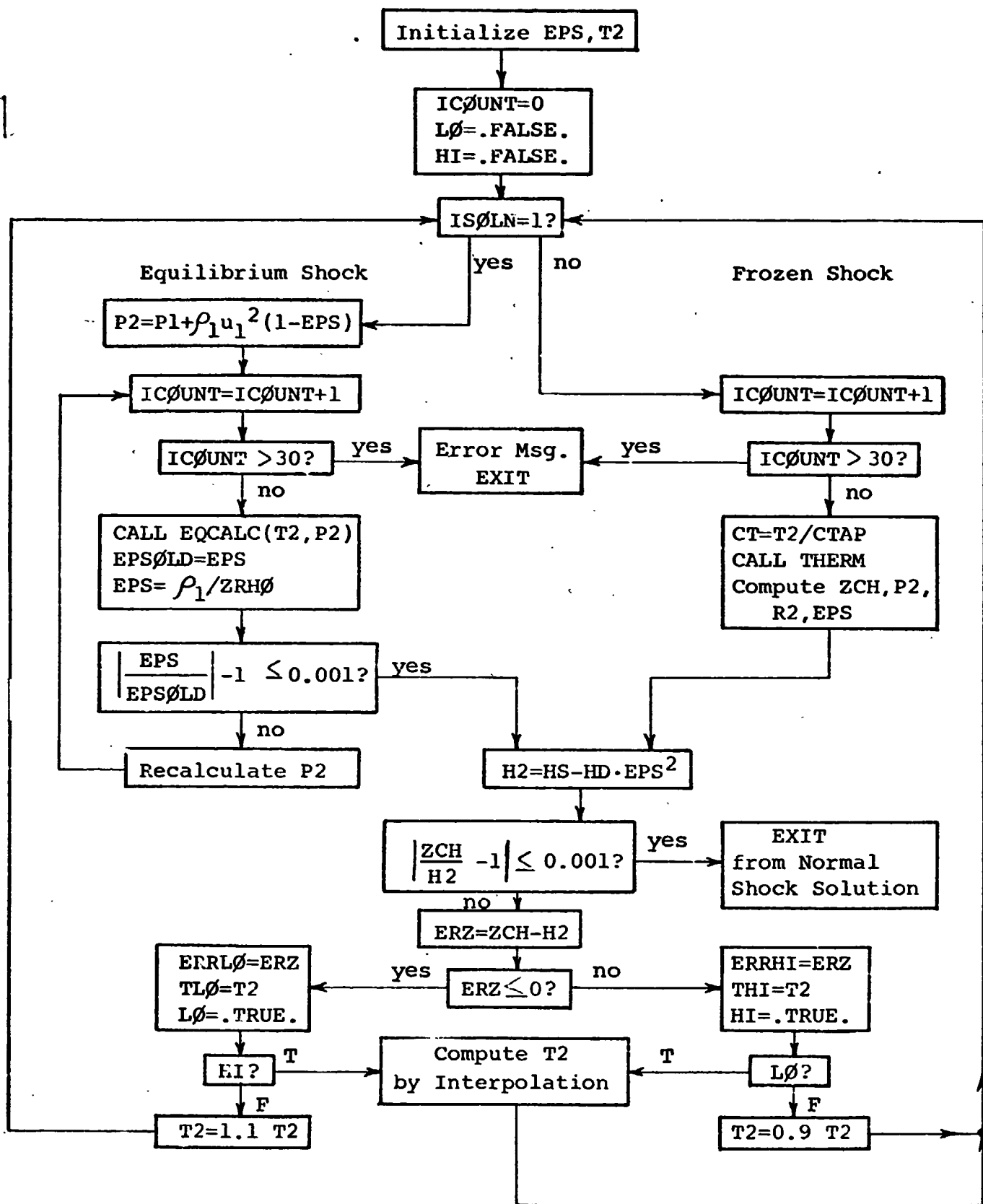
The factor of 1.8 converts the enthalpies from internal code units of cal/gm to Btu/lb; $(30.48)^{3/2}$ converts cm^3/cm^2 to ft^3/ft^2 ; and $(453.6)^{-1}$ converts g to lb.

After the heat fluxes have been computed, the results of the calculations are printed out. To this end, the results are first loaded into the array SCOUT, to allow use of a relatively compact output format. If a binary output tape is being prepared during the run (DATAPE = .TRUE.), certain of the model condition results are also written on tape unit ITPUT.

3.31.2 Normal Shock Solution

Figure 15 diagrams the method used in the normal shock solution. This technique is explained in analytical terms in Section 8.1.1 of Volume I. The solution is based upon an iteration to determine the temperature T2 behind the shock. The value of T2 obtained during the previous call to MDEL is used as a first estimate; this value is stored as T2E (equilibrium shock) or T2F (frozen shock) in common block /STAG/. Prior to the first call to MDEL, these values are initialized to the reservoir temperature.

The criterion for selection of T2 is that the static enthalpy behind the shock, calculated from T2 using equation



ORIGINAL PAGE IS
OF POOR QUALITY
Figure 15. Flowchart of the Normal Shock Solution

I(429b), be equal to the value H2 obtained from the Rankine-Hugoniot energy equation for the shock, equation I(432b). In the case of a frozen shock, the gas is thermally perfect because the mole fractions are constant. The static enthalpy ZCH is calculated, in this case, from species enthalpies obtained by calling subroutine THERM after the nondimensional temperature CT has been set to the value T2/CTAP. The pressure P2 behind the shock is then computed from I(437), and the shock density ratio from I(434).

The equilibrium shock solution is more complicated because in this case the density ratio ϵ is not given as an explicit function of T2. Thus, a double iteration is required. For each trial value of T2, ϵ and the pressure P2 are calculated by an inner iteration as described in the discussion around equation I(438). In this case, the static enthalpy ZCH and density ZRH0 behind the shock are obtained by calling subroutine EQCALC. The outer iteration to determine T2 is the same as for the case of a frozen shock.

The object of the outer iteration is to make $ERZ = ZCH - H2$ equal to zero, to within an accuracy of 0.1 percent. The temperature T2 satisfying this criterion is obtained by iterative linear interpolation:

$$(T2)_{\text{new}} = (T2)_{\text{old}} - ERZ \cdot \frac{THI - TL0}{ERRHI - ERRLO} \quad (114)$$

Here $(T2)_{\text{new}}$ is the improved estimate resulting from the linear interpolation, $(T2)_{\text{old}}$ is the previous estimate, ERZ is the enthalpy error obtained at $(T2)_{\text{old}}$, ERRHI is the last positive error ERZ obtained and THI the corresponding T2 estimate, and ERRLO is the last negative error obtained with TL0 the corresponding temperature. Of course, this formula cannot be used unless points with both positive and negative ERZ have been computed. This requirement is implemented with the aid of logical variables L0 and HI which are initialized to .FALSE.. L0 is set to .TRUE. when a negative ERZ is obtained, and HI is set to .TRUE. when a positive ERZ is found. If ERZ is negative and HI is "false", T2 is increased by 10 percent. If ERZ is positive and L0 is "false", T2 is

decreased by 10 percent. The interpolation formula (114) is not used until both $L\phi$ and HI are .TRUE..

3.31.3 Conditions at the Inviscid Stagnation Point

After the normal shock calculation has converged, the conditions at the inviscid stagnation point are computed as explained in Section 8.1.1 of Volume I. A flowchart of the coding is given in figure 16. The logic is similar to that of the normal shock solution (figure 15), but with two simplifications:

- (1) The stagnation pressure is calculated from the incompressible Bernoulli equation (428b) with no iteration. Thus, only a single iterative loop is required even for the case of equilibrium flow.
- (2) The initial estimate of the stagnation temperature TS is taken to be the temperature $T2$ just behind the normal shock. Since $T2$ is known to be lower than TS , the logical variable $L\phi$ is not needed in the stagnation-condition solution. Linear interpolation is begun as soon as a positive ERZ is obtained, where ERZ is the difference between the known stagnation enthalpy HS and the local enthalpy at the stagnation point (ZCH) computed from TS and PS .

3.32 Subroutine NEWRAP

The function of subroutine NEWRAP is to compute the flow conditions at a point in the equilibrium flow solution, using the method explained in Section 6.2 of Volume I. First, the system of $(c + 1)$ equations I(227) and I(252) is solved for the species mole fractions X_i and the pressure p . Then the remaining flow variables are computed from these data.

The Fortran notation in NEWRAP is similar to that in EQCALC, already discussed in Section 3.18, but there are some differences and additions. ISMCNR denotes the number of dependent species. Initially, this is equal to $n - c = ISS - ISC$.

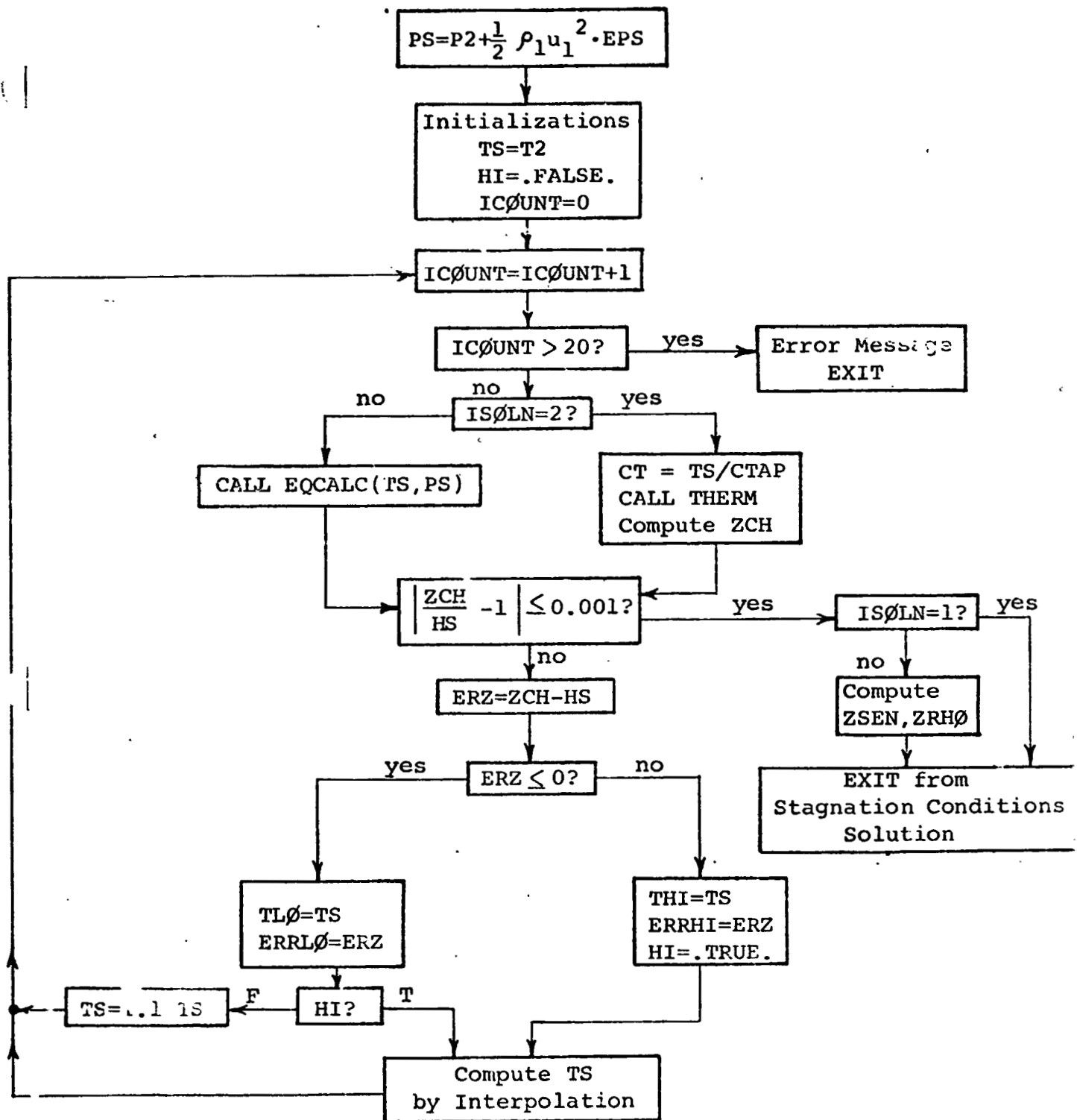


Figure 16. Flowchart of the Solution for Conditions at the Inviscid Stagnation Point

The array CHII is redefined as the logarithm of the equilibrium constants I(225) for the dependent species:

$$\text{CHII}(i-c) = - \frac{\mu_i^0}{R_0 T} + \sum_{j=i_m}^c \bar{\nu}_{i-c,j} \frac{\mu_j^0}{R_0 T} \quad (115)$$

where $i_m = \text{IM}$ is the index of the first independent species, initially equal to 1. A new array PGJ is defined by

$$\text{PGJ}(i) = \frac{S_i^0}{R_0} - \text{CRRB} \cdot W_i \quad (116)$$

where S_i^0 denotes the molar entropy of the i th species at the standard pressure, W_i is the molecular weight of the i th species, and*

$$\text{CRRB} = \frac{S_0}{R_0 W_0} \quad (117)$$

where S_0/R_0 is the nondimensional molar entropy of the gas in the reservoir and W_0 the reservoir molecular weight.

Immediately upon entry, NEWRAP calls subroutine THERM to compute the species thermal properties at the temperature $\text{CT} \cdot T_0$ corresponding to the current value of the nondimensional temperature CT. NEWRAP then sets up the arrays CHII and PGJ, defined above, and obtains a set of initial estimates of the species mole fractions $X_i = \text{CAPX}(I)$ from equation I(1):

$$\text{CAPX}(i) = W \gamma_i \quad (118)$$

The molecular weight $W = \text{CM}$ and species concentrations $\gamma_i = \text{GJ}(i)$ are the values in storage. For the first call to NEWRAP, these are reservoir values. In subsequent calls, they are the final values for the preceding flow point. Also,

*CRRB is set in subroutine INGAS following the entry INTA.

SKIL(I) is initialized to CAPX(I), and ZP to $\ln(\text{PRES})$, where PRES is the stored value of the nondimensional pressure.

The Newton-Raphson iterative loop starts at statement 70. The system of linear equations to be solved in each step of the iteration is I(258). The constant terms $F_j(Y_k^r)$ are loaded into the M2th column of the matrix AA, where $M2 = c + 2$. For $j = 1$ to c , these terms are given by equation I(227), and are computed in the DØ loops ending at statement 90. The constant term for $j = c + 1$ is given by I(252), and is loaded into AA(M1,M2) by the statements between 90 and 100. In the loop "DØ 100", the upper limit ISSNR is the current number of species included, initially equal to ISS.

The matrix of coefficients for the system of equations I(258) is loaded into the first $(c + 1)$ rows and columns of AA. The coefficients I(233b) for $j = i_m$ to c and $k = i_m$ to c go into the first c rows and columns, and are set by the nested DØ loops ending with statement 150. The coefficients I(261b) for $j = c + 1$ and $k = i_m$ to c are computed in the DØ loops ending at statement 170. The coefficient I(261c) for $j = c + 1$ and $k = c + 1$ is set up by the loop DØ 180. Finally, the coefficients I(261a) for $j = i_m$ to c and $k = c + 1$ are loaded into AA by the DØ loops ending at statement 210.

If the number of dependent species ISMCNR is zero a logical indicator NØDEPS is set to .TRUE. at the beginning of NEWRAP. Tests on NØDEPS then cause the subroutine to skip all of the DØ loops with DØ indices running from 1 to ISMCNR, in which sums from $c + 1$ to n are evaluated.

During an equilibrium expansion to high Mach numbers, the mole fractions of ionized species can become extremely low as the gas temperature falls to a small fraction of its value in the reservoir. To avoid computational problems arising when the electron mole fraction becomes exceedingly small, the electrons and all ionized species are eliminated from the equilibrium calculation whenever the electron mole fraction drops below 10^{-20} . The test for this condition is in statement 220. If CAPX(1) is found to be $\leq 10^{-20}$, the following

actions are taken:

- (1) The indicator JJK is reset from its initial value of 0 to 1. The latter value signifies that the elimination of charged species has already been performed.
- (2) The number of ions (IC) is subtracted from ISMCNR and ISSNR.
- (3) IM is increased from 1 to 2.
- (4) AA(1,1) is set to 1, and all the other entries in the first row and first column of AA are set to 0.
- (5) The electron mole fraction, CAPX(1), is set to 0.
- (6) The mole fractions of all the ion species are set to 0.

Regardless of whether the charged species have been eliminated, the system of equations I(258) is solved by calling subroutine DSMSØL. Improved values for the species mole fractions and the pressure are then obtained using equations I(262). The correction factor $1 + h_n^r$ in I(262) is denoted by ZB for $n = 1$ to c (i.e., for the mole fractions), and by ZC for $n = c + 1$ (the pressure). If any of these correction factors turns out to be 0 or negative, the corresponding quantity (mole fraction or pressure) is adjusted by dividing it by 2. Based on the improved values of p and the X_i , the other quantities in the calculation ($ZP = \ln p/p_0$ and $CGMU(K) = \ln X_K$), are then recomputed. Next, the mole fractions for the dependent species are computed from the equilibrium relations I(224), which are written in the logarithmic form

$$\ln X_i = \ln K_{pi} + (\nu_{i-c}^* - 1) \ln p + \sum_{\ell=1}^c \bar{\nu}_{i-c,\ell} X_{\ell} \quad (119)$$

The term $\ln K_{pi}$ is denoted in NEWRAP by CHII(i), equation (115). The $\ln X_i$ are computed in the array SKIL(i-c) and the resulting values are saved in CGMU(i). Then the X_i themselves are computed by taking exponentials and are stored in SKIL(i-c) and in CAPX(i).

The convergence criterion for the iteration is that the absolute magnitudes of all of the relative corrections h_n^r be smaller than or equal to TEST = 10^{-6} . When convergence has been achieved, the other flow variables are computed from the pressure, the mole fractions, and the known reservoir conditions. The relations used in these calculations are equations I(236), I(237), and I(242) for the molecular weight, density, and specific enthalpy, equations I(79b) and I(81) for the effects of gas imperfections, equation I(263) for the flow velocity, and equation I(264) for the mass flux. As noted in Section 3.22.1, the flow variables are computed in nondimensional form.

3.33 Subroutine NEXTMP

The function of subroutine NEXTMP is to determine the location (XMØDEL) of the next point at which model condition calculations are to be done. In the case of channel flow solutions, the routine determines the location corresponding to the next specified channel width.

The first two arguments of NEXTMP specify which of the model points called for in the input have already been passed in the current solution. ITS is the index of the next of the specified test-section diameters or channel widths, while XMØD1 is the x-coordinate the next model point in the geometric sequence of model points.

In the case of channel flow (NPRFLS = 2), the geometric sequence is not used. NEXTMP simply calls subroutine FINDXC to determine the position XMØD2 at which the width of the wide (MBLth) profile has the specified value TCAR(ITS), and sets XMØDEL equal to XMØD2.

In the case of nozzle flow (NPRFLS = 1), the next model point may be either the next point (XMØD1) in the geometric sequence or the point with the next specified area

ratio. In this case, NEXTMP calls FINDX to determine the location (XMØD2) of the point with the area ratio TSAR(ITS), and sets XMØDEL to the lower of the two values XMØD1, XMØD2. If the next point turns out to be at the specified area ratio, ITS is incremented by 1 to signify that TSAR(ITS) has been used. If the next point is a point in the geometric sequence, XMØD1 is replaced by FACMP times XMØD1, which is the next subsequent value of the sequence.

3.34 Subroutine NØNEQ

NØNEQ is the controlling routine for the nonequilibrium flow solution. The overall method of solution has been explained in Section 7.2 of Volume I. Briefly, the solution is started upstream of the throat, near the reservoir, using the perturbation method. At each step in the perturbation solution, the quantities $\xi \chi_i$ (which are measures of the departures of the reactions from equilibrium) are calculated and tested. When any one of these quantities reaches a specified size, the perturbation technique is abandoned and the solution is continued by numerical integration. If the integration is begun well downstream of the throat, it is carried out using the normal, direct method in which the specified nozzle geometry is taken to be a condition on the flow. However, if the integration is started upstream of the throat, an inverse method is used, in which the equilibrium density distribution $\rho(x)$ is used to define the flow. In this case, a switch is made from the inverse method to the direct method when the solution reaches a point slightly beyond the throat.

Figure 17 is an overall flowchart of subroutine NØNEQ. This flowchart shows how the various methods are organized to produce the entire solution. The operations enclosed in double boxes are represented by separate flowcharts and discussed below. IUPD is an indicator which is initialized to 1 (in subroutine READ), and is reset to 0 (in subroutine THRØAT) at the switch from the upstream to the downstream region. INEQ is another indicator, initialized to 0, and set to 1 when the numerical integration is started. Figure 17 does not show any exits from the control loops that generate the solution, because the case termination controls are within the boxes labelled "Numerical integration" and "Perturbation solution". The statements involving MØDLPT, XMSET, and subroutine NEXTMP pertain to the calculation of conditions at model points, and are discussed below.

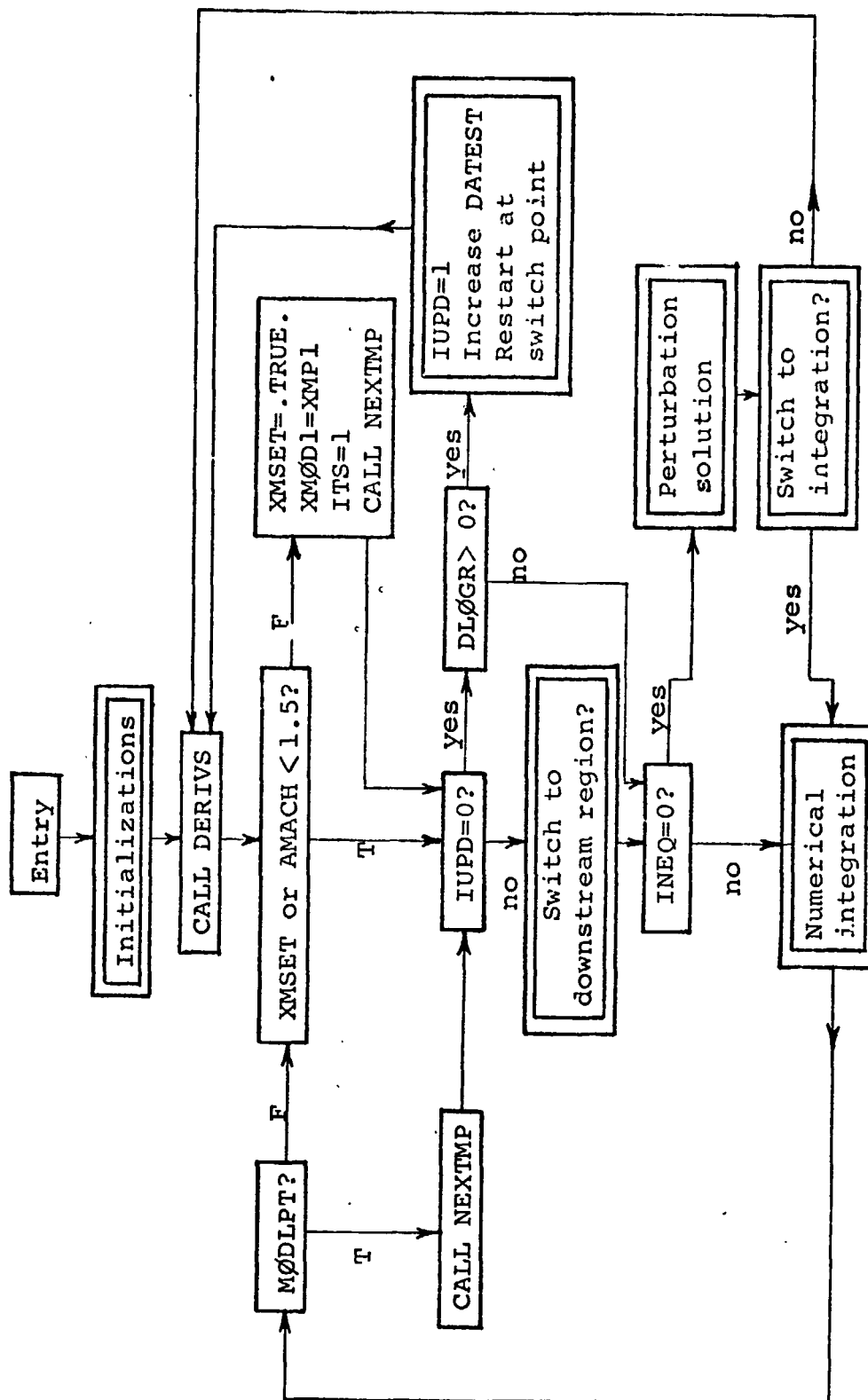


Figure 17. Flowchart of Subroutine NØNEQ

ORIGINAL PAGE IS
OF POOR QUALITY

The remaining parts of this section discuss individual portions and aspects of subroutine NØNEQ in detail.

3.34.1 Initializations

Figure 18 is a flowchart of the initializations section of NØNEQ, represented by the uppermost double box in figure 17. This flowchart shows only the initializations of control variables. In the subroutine, several physical variables are also set, but their values are not needed for following the logic.

The perturbation technique, which is used to start the solution, proceeds by taking steps in temperature, ΔT . The initial temperature decrement is given by $\Delta T/T_0 = \text{DELT1}$, where the variable DELT1 is under input control and has a preset value of 0.01. This preset value of DELT1 gives fairly large steps in the perturbation solution. In cases including the boundary layer ($\text{ISW3B} \neq 0$), to allow more accurate evaluation of the boundary layer integral $I(172)$, DELT1 is reset to $0.049 (1 - \text{CTMAX})$, where CTMAX is the nondimensional sonic temperature T^*/T_0 based on the equilibrium solution. This value gives 20 steps upstream of the throat if the perturbation technique is used all the way to the throat. If the numerical integration is started upstream of the throat, as is usually the case, the number of steps is larger because the initial step size in the integration is much smaller than the initial perturbation step size.

The temperature, species concentrations, and gas properties are initialized for the nonequilibrium solution in the main program. Before calling NØNEQ, the main program sets the nondimensional temperature CT equal to $1. - \text{DELT}$, where DELT is set by the same rules used in calculating DELT1 in NØNEQ. MAIN then calls NEWRAP to determine the equilibrium flow conditions at this temperature, calculates the effective area ratio, and determines the corresponding position in the nozzle by calling FINDX.

As explained in Section 7.3 of Volume I, the system of equations used in the perturbation technique is solvable only if the β_{ij} matrix has a rank of $n-c$. The initializations section of NØNEQ computes the rank IX of $\text{BETA}(I,J)$ and executes an error exit if IX is not equal to $n-c$.

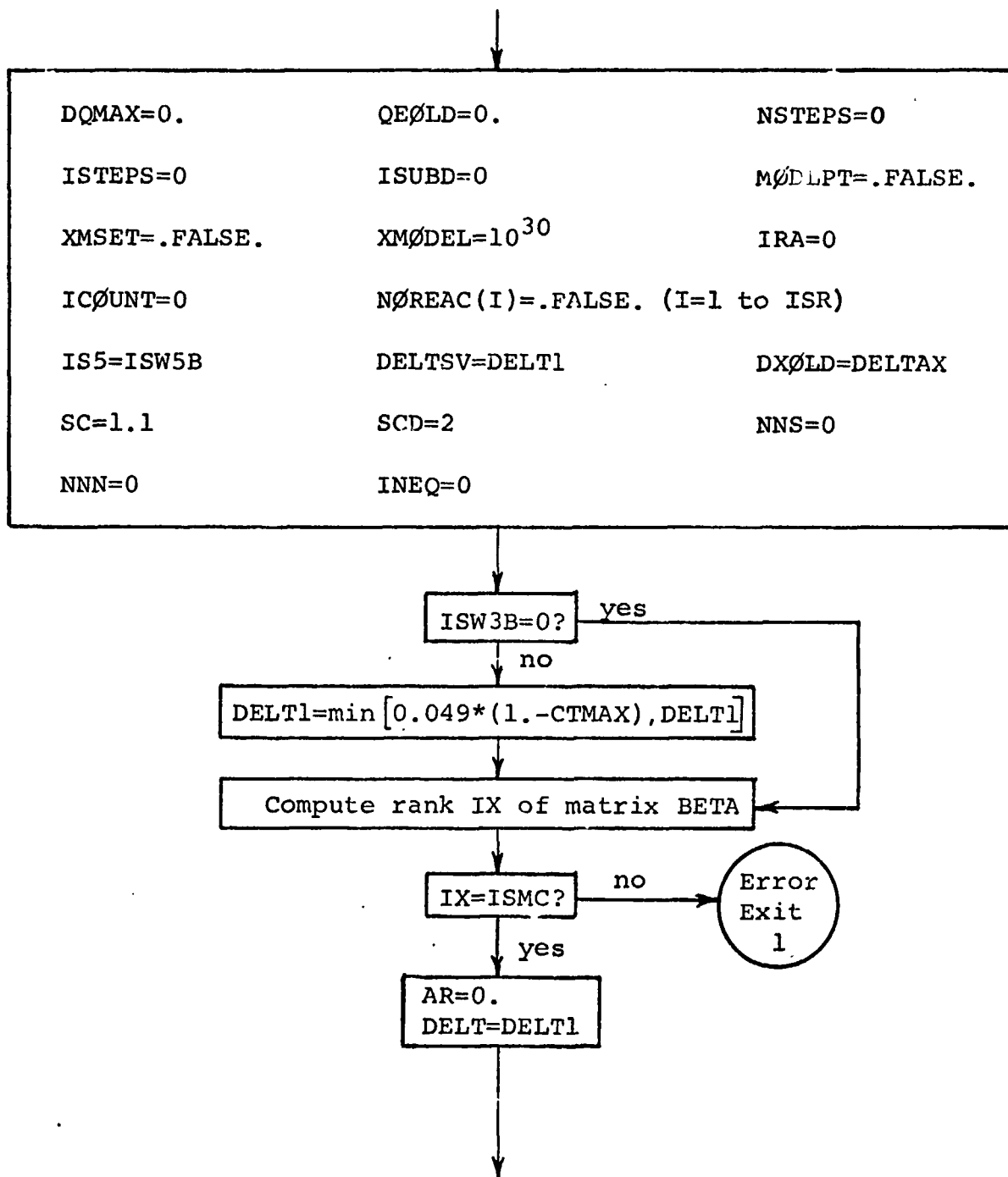


Figure 18. Flowchart of Initializations Section
of Subroutine NØNEQ

3.34.2 Switch to Downstream Region

Figure 19 is a flowchart of the operations represented by the box labelled "Switch to downstream region?" in figure 17. The coding shown in figure 19 determines whether the conditions for switching to the downstream solution have been satisfied, and if so, carries out the switch by calling subroutine THRØAT. The conditions are as follows:

- (1) $DLØGA = d \ln \tilde{A} / dx$, where \tilde{A} is the area ratio computed using the inverse method, must be nonnegative.
- (2) The position of the flow point must lie beyond the geometric throat ($CX > 0$).
- (3) The area ratio $\tilde{A} = AFNX$ must be greater than or equal to a value DATEST which is preset (in subroutine INIT) to 1.01.

When all of these conditions have been satisfied, a large number of flow variable and control parameters are saved in the arrays TB, ITB, and BLBK, and the switch to the downstream region is executed by calling THRØAT. In THRØAT, IUPD is reset to 0 and RSA is set to a value which rescales the actual nozzle geometry to make it consistent with \tilde{A} .

In cases including the boundary layer, the coding in figure 19 also sets the ØMDST(I) array based on the displacement thickness DELBL(I) at the first point where the conditions (1) and (2) are satisfied.

3.34.3 Restart of the Upstream Solution

Figure 20 is a flowchart of the operations represented by the box labelled "IUPD = 1, Increase DATEST, Restart at Switch Point" in figure 17. This section of NØNEQ is entered when a positive value of $d \ln \rho / dx = DLØGR$ is encountered in the downstream solution. If this condition arises so far beyond the upstream/downstream switch point that the effective area ratio $A_0 = AFNX$ exceeds DATEST by 5 percent of DATEST or more, an error exit is executed. Otherwise,

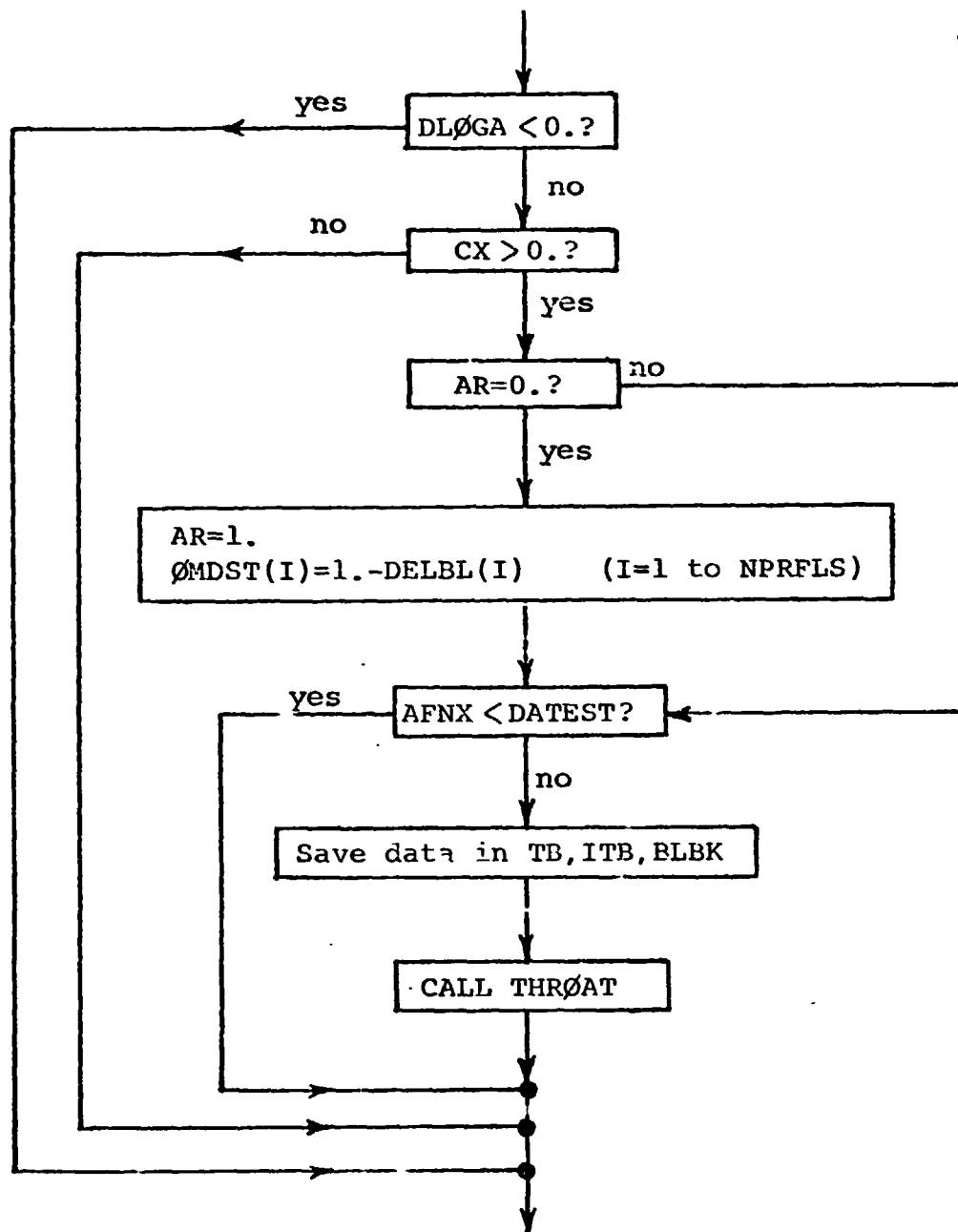


Figure 19. Flowchart of Switch to Downstream Region in Subroutine NØNEQ

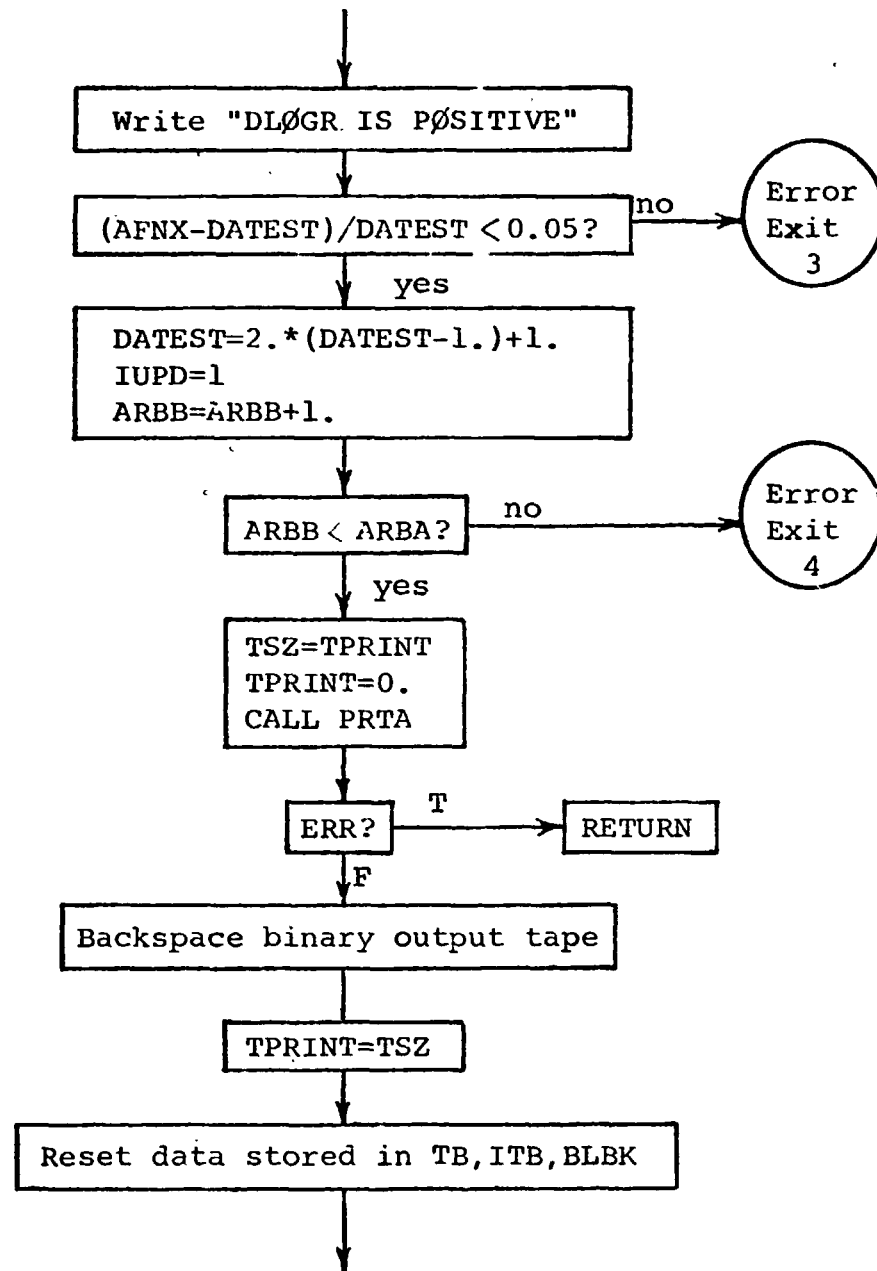


Figure 20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine NØNEQ

the program assumes that the positive $d \ln \rho / dx$ value is a result of switching to the downstream solution too near the throat, and attempts to recover the correct solution. To this end, the excess of DATEST over unity is doubled, IUPD is reset to 1 (signifying the upstream solution), the flow variables and control parameters are all reset to their values at the previous switch point, and the solution is restarted at that point. A counter ARBB (initialized to 0. in subroutine INIT) is also incremented by 1. If the $d \ln \rho / dx > 0$ condition is encountered repeatedly after successive restarts, the effort to continue the solution is abandoned after four attempts ($ARBB \geq ARBA = 5.$, set in INIT). Also, before the restart is executed, the conditions at the final point of the invalid downstream solution are printed out by setting TPRINT = 0. and calling PRTA. If a binary output tape is being produced for subsequent use by the NATA plot program, the records containing data from the invalid solution beyond the switch point are eliminated by backspacing the tape.

3.34.4 Perturbation Solution and Switch to Integration

Figure 21 is a flowchart of the operations represented by the boxes labelled "Perturbation solution" and "Switch to integration?" in figure 17. These operations may be summarized as follows:

- (1) Subroutine PERT is called to compute the perturbations PCT (in the nondimensional temperature), PERTGJ(J) (in the species concentrations), and SDCHI(I) (in the reaction parameters χ_i).
- (2) If the solution is in the downstream region (IUPD = 0) and the nondimensional temperature decrement DELT is still equal to the reduced value of DELT1 (see part 3.34.1 above), DELT is reset to the input value of DELT1, which was saved in DELTSV. This is done because the step size in the perturbation solution is sufficiently small in the throat and near downstream region, even when the larger decrement is used.

[illegible]

Figure 21. Flowchart of Perturbation Solution and Switch to Integration in Subroutine N0N2Q

- (3) The largest $SDCHI(I)$ value ($DCHMAX$) and the smallest ($DCHMIN$) are determined.
- (4) The ratio $RATIO = DCHMIN/DCHMAX$ is examined. If this ratio is larger than or equal to a value $DCHRAT$ (preset in $BLCK DATA$ to 10^{-4}), $NONEQ$ proceeds to the checks for the switch to numerical integration, item (5) below. If $RATIO$ is less than $DCHRAT$, the reaction system assumed in the gas model will lead to a switch to integration at a point where some of the reactions are still very near equilibrium. The value of $RATIO$ is approximately independent of the size of $DCHMAX$; thus, the program responds to an excessively small value of $RATIO$ immediately, even though $DCHMAX$ may be much smaller or much larger than the value $CCHI$ at which the switch to integration is desired. To prevent the premature start of the numerical integration, $NATA$ artificially increases the rate constant for the reaction ($I = I_{MAX}$) for which $|SDCHI(I)| = DCHMAX$, if this can be done without significantly affecting the solution. This procedure is discussed near the end of Section 7.3 in Volume I. If $RATIO < DCHRAT$, $NONEQ$ determines whether this reaction involves a minor species (concentration less than or equal to $GAMIN$, preset to 10^{-10}) whose concentration is being decreased by the reaction. If so, the rate constant for the I_{MAX}^{th} reaction is increased by a factor of $RATIO^2 = 1.1 * DCHRAT / RATIO$, and the perturbation calculation for the previously assumed temperature is repeated. If not, the perturbation solution is continued without adjusting the rate constant.
- (5) The next step is to check whether $DCHMAX$ has reached the value $CCHI$ at which the switch to the numerical integration is desired. If $DCHMAX$ is less than $CCHI$, the perturbation solution continues. If the axial coordinate

CX is greater than or equal to the current value of XMØDEL (the coordinate of the next model point), the indicator MØDLPT is set to .TRUE., and subroutine NEXTMP is called to determine a new value for XMØDEL. In any event, subroutine PRTA is called to provide output of the computed flow conditions at the current point of the perturbation solution. If MØDLPT is .TRUE., PRTA also calls subroutine MØDEL to calculate test conditions on a model.

- (6) After the return from PRTA, the case termination tests are applied. If CX is greater than or equal to CXMAX, or if the nondimensional temperature CT is less than or equal to TSTØP, control is transferred to statement 660. The flow conditions and model conditions at the final point are then printed and the RETURN is executed.
- (7) If case termination has not been reached, the temperature CT is decremented by DELT, and subroutine NEWRAP is called to compute the equilibrium flow conditions at the new temperature. The equilibrium effective area ratio AFNTS is then calculated from the continuity equation I(265), and subroutine AXFIT is called to determine the axial coordinate CX at this area ratio. Control is then transferred to statement 220 ("CALL DERIVS" in figure 17).
- (8) Once DCHMAX reaches or exceeds the criterion value CCHI for the switch to integration in step (5), NØNEQ commences an iterative adjustment of the nondimensional temperature CT in order to obtain a flow point at which $DCHMAX = |\delta \chi_i|_{\max}$ satisfies both of the conditions in equation I(381). Usually the perturbation solution overshoots the second condition I(381), i.e., DCHMAX is greater than PCTEST*CCHI, where

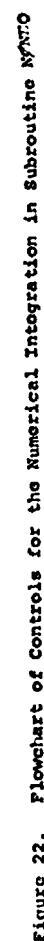
PCTEST = 1.2 (set in subroutine INIT). When this occurs, NØNEQ cuts DELT in half, adds DELT to CT in order to obtain a temperature halfway between the current value and the preceding one, and computes the corresponding point in the perturbation flow solution by calling NEWRAP and AXFIT as in step (7) above. This procedure, together with the normal procedure for advancing the perturbation solution (step (7)), repeatedly subdivides the temperature interval known to contain the flow points satisfying the conditions I(381).

- (9) When a point satisfying both of the conditions I(381) has been obtained, the switch from the perturbation technique to numerical integration is carried out. The indicator INEQ is reset from 0 to 1 and the perturbations PCT, PERTGJ(J) and PRHØ are added to the corresponding equilibrium flow quantities CT, GJ(J), and RHØ. The integration step size DELTAX is adjusted, if necessary, to make it no larger than $DXM = 0.01 * DCHMIN/DCHLL$, where the input DCHLL is preset to 10^{-4} in BLOCK DATA. This adjustment provides a reduced initial step size when $DCHMIN = |\delta \chi_i|_{\min}$ at the switch point is smaller than 10^{-4} , i.e., when some of the reactions are still very close to equilibrium. If this reduction in DELTAX were omitted, the normal step size controls of the integration would still bring DELTAX down to similar values at the cost of some additional computation. Any rate constants which have been artificially increased in step (4) are now restored to their correct values. Subroutine DERIVS is called to compute the derivatives and supplementary flow variables at the switch point, based on the perturbed temperature and concentrations, and control is transferred to the numerical integration section of NØNEQ. If the indicator FAILED is .TRUE. after the call to DERIVS, an error exit is executed. This indicator is .TRUE. if the flow conditions fail to pass certain validity checks in DERIVS and CØMM.

3.34.5 Numerical Integration

Figure 22 is a flowchart of the operations represented by the box labelled "Numerical integration" in figure 17. These operations may be summarized as follows:

- (1) The case-termination checks are the same as in the perturbation solution. If $CX \geq CXMAX$ or $CT \leq TSTOP$, control is transferred to statement 660. The flow conditions and model conditions at the final flow point are then printed, and the RETURN is executed.
- (2) If the end of the case has not been reached, the counter NNN is incremented and subroutine PRTA is called to provide output of the conditions at the flow point reached during the preceding integration step.
- (3) Next, the integration step size DELTAX and the factors SC and SCD used in changing DELTAX are adjusted. If NNN is equal to NQS (an input parameter, preset to 4), then DELTAX is multiplied by SC, NNS is incremented by 1, and NNN is reset to 0. Then if NNS is equal to NQS, SC is increased by 0.1, SCD is reset to the new SC value, and NNS is reset to zero. For $NNN < NQS$, no changes are made in these variables.
- (4) In preparation for the next integration step, the current values for CX, CT, GJ(I), and other quantities are saved in the storage locations CXB, CTB, GJB(I), etc. If the forthcoming step should fail for any reason, these data will be used in restarting the solution at the beginning of the step with a smaller step size.
- (5) Next, the anticipated position, $CX + DELTAX$, at the end of the forthcoming step is compared with $XMODEL$ and $CXMAX$. If $CX + DELTAX$ is greater than either of these values, DELTAX is reduced to make the end of the step coincide with the lesser of



ORIGINAL PAGE IS
OF POOR QUALITY

(1) the two. If the end of the step will be at XMØDEL, then MØDLPT is set to .TRUE. and the DELTAX value before the reduction is saved in DXSAVE.

- (6) Subroutine RNKT is called to compute the changes in the dependent variables over the integration step.
- (7) The indicator FAILED is tested. A .TRUE. value of FAILED indicates that a validity check was not satisfied somewhere in RNKT, DERIVS, or CØMM. If FAILED = .TRUE., the step size is reduced by dividing DELTAX by SCD and the calculation is restarted at the beginning of the step, using the data saved in step (4). In the event of such a step restart, a counter ICØUNT is incremented. Also, MØDLPT is set to .FALSE., SCD is increased by a factor of 1.1, and SC is reduced by 0.1 (but not below a value of 1.1). In addition, NNN and NNS are both set to zero. The idea underlying these changes in the step control parameters is that the failure was caused by instability resulting from the use of an excessively large step size. Either the step has been increased too rapidly, or the solution is entering a region where the reaction system requires a smaller step size for stability. Thus, the step size itself is reduced, and the growth in step size is both temporarily arrested and reduced in rate. The increase in SCD allows the program to reduce the step size very rapidly if repeated failures are encountered in the same step, as sometimes happens in runs using electronic nonequilibrium models.
- (8) If FAILED is .FALSE. after the call to RNKT, additional validity checks I(407) and I(408) are applied to the results of the integration step. If any of these conditions is violated, the step is restarted with a smaller step size as described above in item (7). Also an indicator IFAIL is set to a distinctive value for each possible type

of failure. IFAIL is also set when step failures occur in other subroutines. Each time a step fails and is restarted, the IFAIL value is saved in the integer array IF(ICOUNT). These values can be obtained in the printed output by setting ISW5B to a negative value in the code input. Output of these data shows what tests, species, etc., are controlling the integration step size.* If a failure occurs in one of the conditions I(407a), I(407c) on the species concentrations, then the program checks whether the procedure of freezing the concentration of a minor species (Section 7.5.3 of Volume I) is applicable. The conditions for use of this procedure are that the species I, whose concentration GJ(I) violates I(407a) or I(407c), should have a concentration GJB(I) at the start of the step less than GAMIN, and that the concentration change for this species over the step, SDGJ(I), be negative. If these conditions are satisfied, all of the reactions contributing to changes in the concentration of this species are frozen by setting the corresponding elements of the array NØREAC(K) to .TRUE.. The step is then restarted in the usual way.

- (9) If the conditions I(407) and I(408a,b) are satisfied, the integration step is presumed to be valid subject to the condition I(408c), which is applied later. The changes SDT, SDTE, SDCHA in the dependent variables during the step are added to the values CTB, CTEB, CHB at the start of the step to obtain the corresponding values at the end of the step. The species concentrations are then adjusted to restore the original elemental composition of the gas mixture, as described in Section 7.1.3 of Volume I and in Section 3.34.6 below. Subroutine DERIVS is then called to compute the derivatives and supplementary flow variables at the end of the step.

*The numerical code used in IF(ICOUNT) is defined in Section 4.55.

- (10) If $NT = 2$ (electronic nonequilibrium model), the condition I(408c) on the change in the energy transfer to the electron gas (QDPE or \dot{q}_e) is now applied to the final QDPE value resulting from this call to DERIVS. If this condition is not satisfied, the step is restarted with a reduced step size in the usual way.
- (11) Once all of the validity checks have been passed, SCD is reset equal to SC and the step-restart counter ICOUNT is reinitialized to 0. The counter ISTEPS for successful steps is incremented by 1. The step size DELTAX is set equal to the larger of DELTAX itself and $0.7 \cdot DXOLD$. Thus, if DELTAX has been reduced sharply by repeated step restarts, it is here returned to 70 percent of its value in the preceding step.
- (12) Control is now transferred to statement 210 ("MODLPT?" in figure 17).

3.34.6 Element Conservation

The adjustment of species concentrations to restore conservation of the chemical elements, explained in Section 7.1.3 of Volume I, is implemented by the coding from statement 800 down to 940 in NONEQ. The programming is straightforward, and can easily be related to the formulas in Section 7.1.3 with the aid of the following summary of notations:

CJ	c_j'
DCI(J)	δc_j
GSQ(I)	$(\gamma_i')^2$
AIN(J,K)	A_{jk}
DCA(K)	$\frac{1}{(\gamma_k')^2} \sum_{j=1}^C \delta c_j \cdot A_{jk}$
AA(I,ISMCP1)	b_i (before call to DSMSOL)

CDIJ(I,K)	$\bar{v}_{i-c,k}$
AA(I,L)	a_{il}
AA(I,ISMCP1)	$\delta \gamma_{c+1}$ (after call to DSMSØL)

The system of equations I(346) is solved for $\delta \gamma_m$ by calling subroutine DSMSØL. After the adjusted concentrations of the dependent species ($i = c + 1$ to n) have been calculated from I(340) and those of the independent species ($i = 1$ to c) from I(338), the condition I(407a) is applied to the adjusted concentrations. If any of these concentrations is found to be negative, the step is restarted with a reduced step size in the usual way.

3.34.7 Model Points

The position of the next model point, at any point in the flow, is denoted by XMØDEL. Whenever a particular model point has been reached and the model calculations have been done, the new XMØDEL value is obtained by calling subroutine NEXTMP(ITS,XMØD1,XMØDEL). The argument ITS is the index of the next specified test section diameter, and XMØD1 is the coordinate of the next model point in the geometric sequence of model points. Subroutine NEXTMP, each time it is called, determines whether the next model point is at the test section diameter with index ITS or at XMØD1, sets XMØDEL accordingly, and updates ITS or XMØD1 as required. Before NEXTMP is called for the first time in NØNEQ, the initial values ITS = 1 and XMØD1 = XMP1 must be set. This is done by coding shown in figure 17, at the first point where the Mach number AMACH has reached or exceeded a value of 1.5. A .TRUE. value of the indicator XMSET denotes that this initialization has already been performed.

The indicator MØDLPT, if .TRUE., denotes that the current flow point in the perturbation solution or the integration is a model point. If MØDLPT = .TRUE., PPTA calls entry ØUT2 of subroutine ØUT1 to print the flow conditions (even though the TPRINT criterion may not be satisfied) and calls MØDEL to compute and print the model test conditions.

When a model point is encountered during the numerical integration, DELTAX is reset to provide a flow point exactly at the model point. During the perturbation solution, MØDEL is called at the first solution point beyond XMØDEL. To obtain a perturbation solution point coinciding with XMØDEL would require an iteration, as in the equilibrium and frozen flow solutions, because the flow is generated by decrementing the temperature, not by incrementing the position coordinate. Such an iteration has not been provided in the coding because, in the normal applications of NATA, the switch from the perturbation technique to the numerical integration always occurs upstream of the point where the Mach number is equal to 1.5.

3.34.8 Error Exits

Subroutine NØNEQ contains ten error exits, marked conspicuously with comment cards in the listing. Whenever one of these exits is executed, a message giving the error exit number is written out before the DUMP routine is called to provide diagnostic data. This message facilitates determining the nature of the error that caused NØNEQ to fail.

3.35 Subroutine NRMAX

This subroutine determines the sonic flow conditions for the case of equilibrium flow. Its logic and structure are practically identical with those for the corresponding calculation for frozen flow, performed in subroutine FRØZEQ. The flowchart for the frozen calculation (figure 9) is applicable to subroutine NRMAX. The discussion in Section 3.22.2 is also applicable. In the present case, the mass flux at each temperature is determined by calling subroutine NEWRAP.

3.36 Subroutine ØUT

This routine simply prints a list of definitions of the labels used in the output of the flow conditions and test conditions on models and wedges.

3.37 Subroutine \emptyset UT1

Subroutine \emptyset UT1 is the principal output routine of NATA. When called through its first entry point, it produces a printed list of the conditions at a special point in the flow, such as the reservoir or the sonic point. When called through its entry \emptyset UT2(IS \emptyset LN), it prints a more comprehensive specification of the conditions at a general flow point. In addition, if the indicator DATAPE is .TRUE., a call to \emptyset UT2 causes a record to be written on the binary output tape (tape 8), for subsequent use by the NATA plot program. Figure 23 is an overall flowchart of the subroutine.

3.37.1 Entry \emptyset UT1

\emptyset UT1 is called (by MAIN) to print the conditions in the reservoir and at the frozen and equilibrium sonic points. Since a call to \emptyset UT1 precedes each of the three flow solutions, \emptyset UT1 is also used to initialize some quantities used by \emptyset UT2.

The chemically frozen specific heat ratio GAMMA is computed in a section of coding which is used by both \emptyset UT1 and \emptyset UT2. First, the entry THERM1 of subroutine THERM is called to compute the nondimensional species specific heats $C_{pi}/R_0 = CCPJ(I)$ at the current nondimensional temperature $CT = T/T_0$. The nondimensional molar specific heat of the gas mixture is then computed as

$$C_P \equiv \frac{C_p}{R_0} = \sum_{i=1}^n X_i \frac{C_{pi}}{R_0} \quad (120)$$

where the species mole fractions X_i are obtained from the array SAVEC(I), where they have been placed by the calling routine. Then the thermodynamic relation

$$C_p - C_v = R_0 \quad (121)$$

is used to obtain the specific heat ratio, γ :

$$\gamma = \frac{C_p}{C_v} = \frac{1}{1 - \frac{R_0}{C_p}} \quad (122)$$

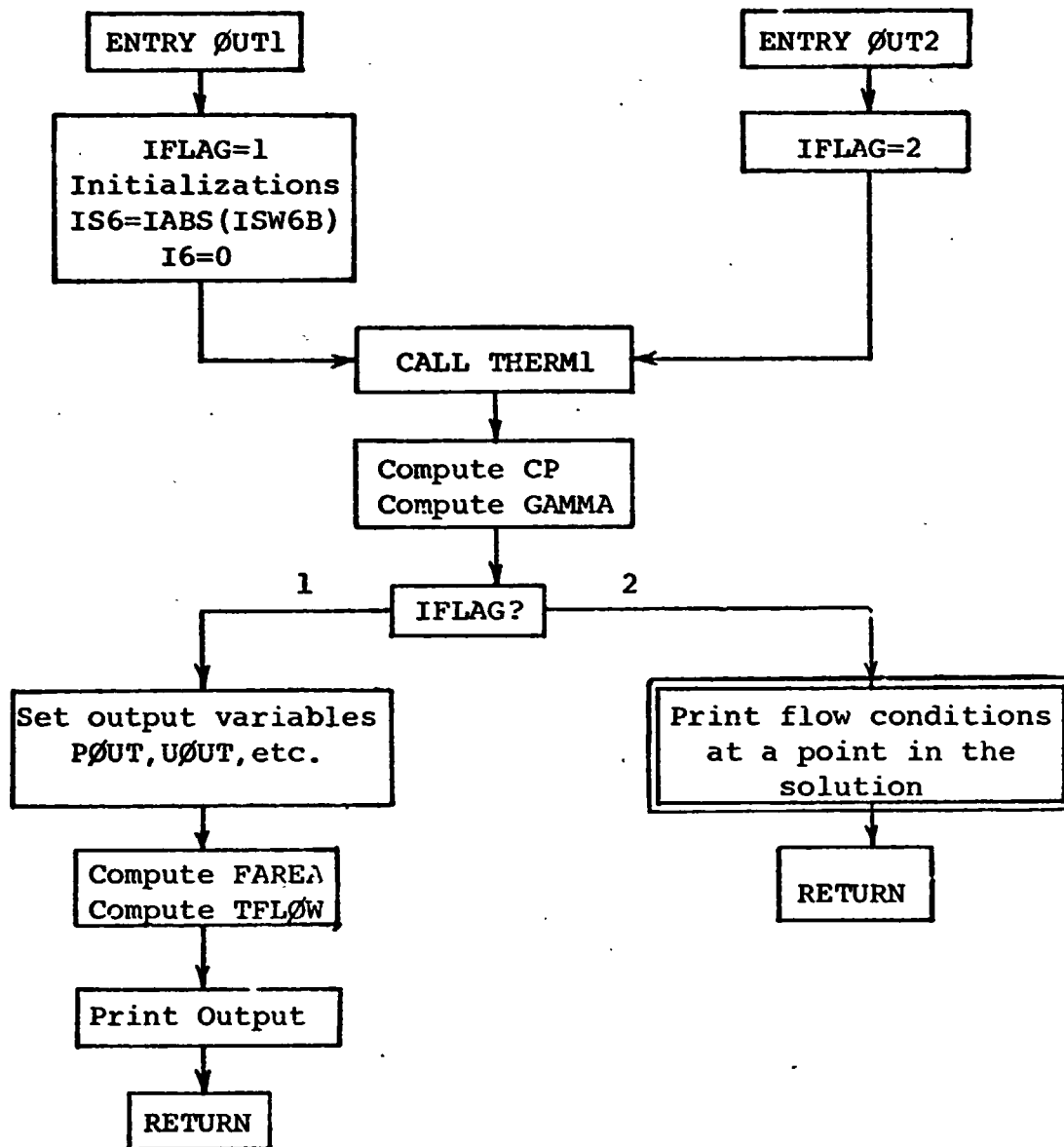


Figure 23. Overall Flowchart of Subroutine ØUT1

At the conclusion of this calculation, the indicator IFLAG is used to transfer control to the appropriate portion of the subroutine, depending on which entry point was called. If entry was through ØUT1, control passes to statement 30. The variables used for output of the flow conditions in ØUT1, namely PØUT, UØUT, etc., are then set. Next, the effective cross sectional area of the flow at the throat, FAREA, is computed. For ISW3B = 0 (boundary layer neglected), FAREA is based upon the geometric dimensions of the nozzle or channel at the throat. For ISW3B ≠ 0, the displacement thickness at the throat is taken into account using equation I(125), I(129), or I(133) as appropriate. After it has been calculated in square centimeters, FAREA is converted into square feet. The total mass flow is then computed by multiplying FAREA by the sonic mass flux in lbm/ft²sec.

Finally, the flow conditions, including the calculated mass flow TFLØW, are printed out in the form of a list. This is followed by a list of species mole fraction.

3.37.2 Entry ØUT2

The operations performed when the entry ØUT2(ISØLN) is called, which are represented by a double box in figure 23, are flowcharted in detail in figure 24. The complexity of this portion of the subroutine results from the need to deal with various special cases:

ISW3B ≠ 0	Boundary layer included; output includes boundary layer displacement and momentum thicknesses, heat flux, shear stress, Stanton number, etc.
ISW3B = 0	Boundary layer neglected.
NPRFLS = 1	Flow is in a nozzle.
NPRFLS = 2	Flow is in a channel; two full sets of boundary layer outputs if ISW3B ≠ 0.
JDIM = 0	Two dimensional nozzle.

ORIGINAL PAGE IS
OF POOR QUALITY

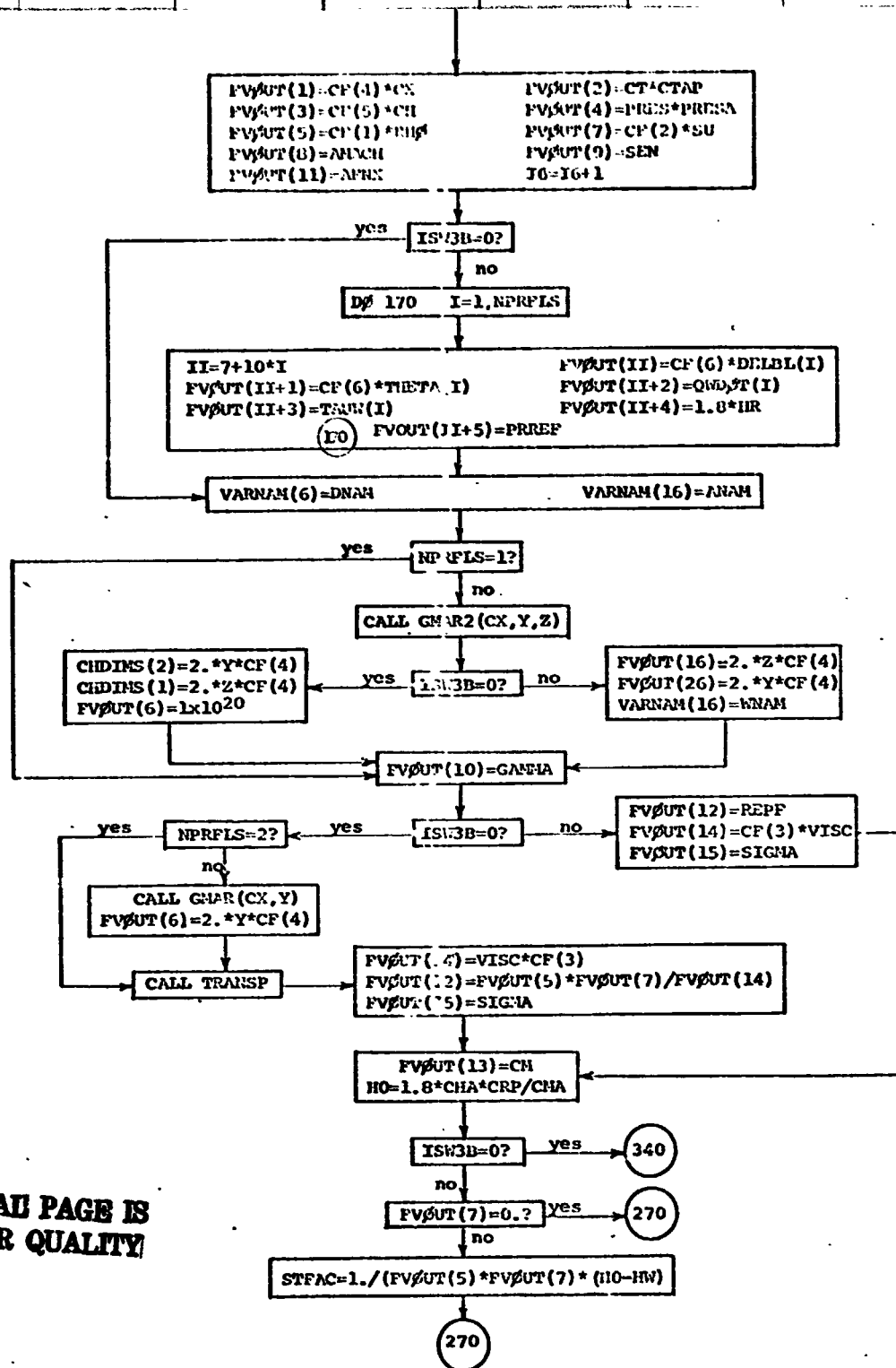
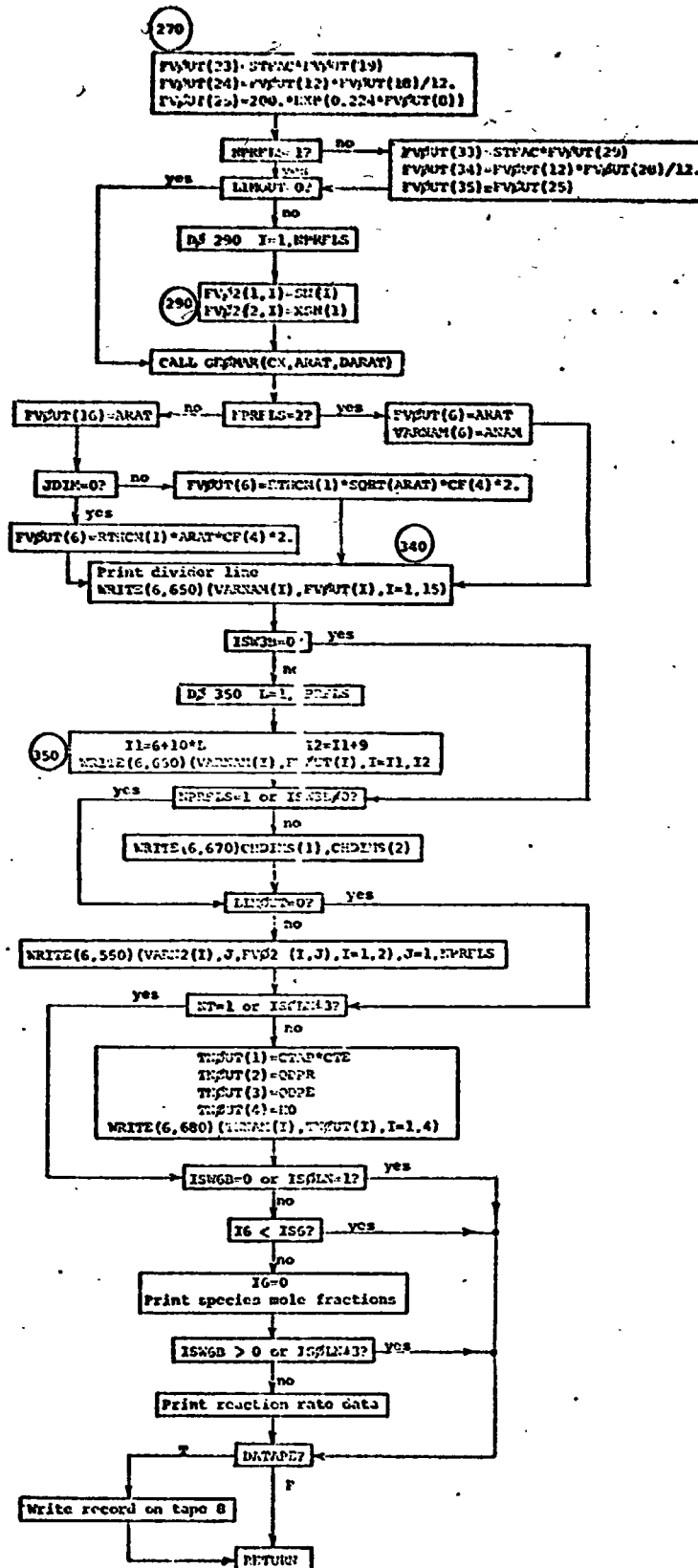


Figure 24a. Flowchart of the Operations Produced by Calling Entry OUT2 of Subroutine OUT1 (Part a)



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 24b. Flowchart of the Operations Produced by Calling Entry PUT2 of Subroutine PUT1 (Part b)

JDIM = 1 Axisymmetric nozzle.

LIMOUT \neq 0 Include output of SN, XSN boundary layer parameters.

LIMOUT = 0 Omit this output.

The printed outputs are, in most cases, expressed in different physical units from those used in the internal computations of the code. Thus, a major function of \emptyset OUT2 is to re-express these quantities in the desired output units. The converted flow variables in output units are loaded into an array FV \emptyset UT(I), dimensioned (35). This scheme allows the use of compact WRITE statements based on implied DO loops. The converted quantities in FV \emptyset UT are also used in the calculations of conditions on wedge models; they are communicated to subroutine WEDGE through the common block \emptyset UTPUT. Definitions of the FV \emptyset UT array elements are given in the glossary of Fortran symbols, Section 4.31. However, the definitions will be reviewed here to facilitate the study of the flowchart (figure 24).

The following elements of FV \emptyset UT always have the meanings indicated:

- (1) Axial coordinate, x (inches)
- (2) Temperature, T ($^{\circ}$ K)
- (3) Enthalpy, h (Btu/lb)
- (4) Pressure, p (atm)
- (5) Density, ρ (lbm/ft³)
- (7) Velocity, u (ft/sec)
- (8) Mach number, M
- (9) Entropy, s (Btu/lbm- $^{\circ}$ R)
- (10) Frozen specific heat ratio, γ

- (11) Effective area ratio, A_e
- (12) Reynolds number per foot
- (13) Molecular weight (g/mole)
- (14) Viscosity (lbm/ ft-sec)
- (15) Electrical conductivity (mho/cm)

Element number 6 is absent from this list because its meaning varies with the type of case, as explained below. For $ISW3B \neq 0$, the Reynolds number per foot, viscosity, and electrical conductivity are calculated in BLAYER, and ØUT2 obtains these data from common blocks /BLØUT/ and /TRPRØP/. For $ISW3B = 0$, BLAYER is not called and these quantities must be computed in ØUT2. The viscosity and conductivity are obtained by calling subroutine TRANSP.

If the boundary layer is included ($ISW3B \neq 0$), the ten elements FVØUT(16) to FVØUT(25) are loaded with data relating to the boundary layer. If, in addition, a channel geometry is being used ($NPRFLS = 2$), these elements contain the data for the boundary layer on the first profile, and the ten elements FVØUT(26) to FVØUT(35) contain the corresponding data for the second profile. The meanings for the elements numbered 17 to 25 are as follows:

- (17) Boundary layer displacement thickness (inch)
- (18) Boundary layer momentum thickness (inch)
- (19) Heat flux to nozzle wall (Btu/ft²-sec)
- (20) Shear stress on nozzle wall (lbf/ft²)
- (21) Recovery enthalpy (Btu/lbm)
- (22) Prandtl number at the reference temperature
- (23) Stanton number

(24) Reynolds number Re_0 based on momentum thickness

(25) Value of Re_0 for boundary layer transition

The data on geometric area ratio and nozzle diameter or channel widths is placed into FVØUT(6), FVØUT(16), and FVØUT(26). The arrangement of these data depends upon the type of case being treated:

- A. If ISW3B = 0, NPRFLS = 1: FVØUT(6) = nozzle diameter (inch). In this case, the geometric area ratio is equal to the effective area ratio, FVØUT(11).
- B. If ISW3B = 0, NPRFLS = 2: FVØUT(6) is set to a large value which exceeds the width of the F format used and thus prints as a row of asterisks. The width of the channel surface lying on the first profile is set to CHDIMS(2). The corresponding width for the surface with the second profile goes into CHDIMS(1). These CHDIMS(I) values are printed on a separate line in the block of output. The geometric area ratio is again equal to the effective area ratio.
- C. If ISW3B \neq 0, NPRFLS = 1: FVØUT(16) = geometric area ratio, FVØUT(6) = nozzle diameter (inch).
- D. If ISW3B \neq 0, NPRFLS = 2: FVØUT(6) = geometric area ratio; FVØUT(16) = width of channel surface lying on the first profile (inch); and FVØUT(26) = width of channel surface lying on the second profile (inch).

Note that in the channel case (NPRFLS = 2), the width of the surface lying on a given profile is twice the ordinate of the other profile.

After the data have been loaded into the output arrays, they are printed out. First, a line of asterisks is printed to separate the new block of output from the previous one. Imbedded in this line is a word ("frozen", "equilibrium", or "nonequilibrium") identifying the type of solution being

computed, based upon the entry-point argument ISØLN. In the case of a nonequilibrium solution, the number of integration steps since the last output of flow variables (NSTEPS) and the value of the indicator INEQ are also printed in this divider line. Note that NSTEPS is initialized in ØUT1 to the seven-digit value 1000000, so that when printed (with an I6 format) in the frozen and equilibrium solutions, it gives a row of six asterisks.

Following the divider line, the first fifteen elements of FVØUT(I) are printed by a single WRITE statement. The array VARNAM(I) appearing in this statement contains the Hollerith labels for the output variables. Then, if ISW3B is nonzero, the boundary layer data in FVØUT(16) to FVØUT(25) or FVØUT(35) are printed. For channel cases without the boundary layer, the channel widths CHDIMS(I) are printed on a separate line. If LIMØUT is nonzero, the boundary layer parameters stored in FVØ2(I,J) are printed on a separate line. For nonequilibrium solutions based on an electronic nonequilibrium model, the electron temperature, the radiative and electronic energy transfer terms, and the total enthalpy are loaded into TNØUT(I) and printed.

Next, if ISW6B is nonzero and ISØLN is not equal to 1, and if I6 is equal to IS6, the species mole fractions are printed. The test on I6 provides output of these species data every IS6th time ØUT2 is called in the equilibrium and nonequilibrium solutions, where $IS6 = |ISW6B|$. Finally, if ISW6B is negative and ISØLN = 3 (nonequilibrium solution) and if I6 = IS6, a table of reaction rate data is printed out. This table includes P_i , χ_i , and $P_i \chi_i$ for all of the reactions ($i = 1$ to r), and $d \ln \gamma_j / dx$ for all the species ($j = 1$ to n).

If DATAPE is .TRUE., a record is written on tape 8 for subsequent use by the NATA plot program.

3.38 Subroutine PERT

The function of subroutine PERT is to compute the perturbations in the flow variables by solving the following system of $n+2$ linear equations:

Equation IndexEquation

1 to c	I(368) for j = 1 to c
c+1 to n	I(362) for j = c+1 to n
n+1	I(374)
n+2	I(380)

The unknowns are $\text{PERTGJ}(J) = \delta \chi_j$ for $j = 1$ to n , $\text{PCT} = \delta T/T_0$ and $\text{PRH}\emptyset = \delta \rho / \rho_0$, in the order^j listed. The simultaneous solution is obtained by calling subroutine $\text{DSMS}\emptyset\text{L}$.

The coefficients and constants for the above system of equations are set up in the common array AA. First, the region of AA which is to be used is zeroed. Then, in the $\text{D}\emptyset$ loops ending at statement 20, the coefficients for equations I(368) are set into $\text{AA}(\text{I},\text{J})$ for $\text{I} = 1$ to ISC . The coding down to the statement following 110 then loads the coefficients of equations I(362) into $\text{AA}(\text{I},\text{J})$ for $\text{I} = \text{ISC} + 1$ to ISS , and loads the coefficients for equation I(374) into $\text{AA}(\text{ISSP1},\text{J})$. In the calculation of the coefficients for I(362), the Fortran index I corresponds to j in I(362), J corresponds to k and K corresponds to i. The coding from $\text{D}\emptyset$ 120 down to the second statement below 130 sets up the coefficients for equation I(380) in $\text{AA}(\text{ISSP2},\text{I})$. Finally, the loop $\text{D}\emptyset$ 140 loads the constant terms of equation I(362) into $\text{AA}(\text{I},\text{ISSP3})$ for $\text{I} = \text{ISCP1}$ to ISS . The remaining equations are all homogeneous, so that their constant terms have been set by the initial zeroing of the AA array.

After $\text{DSMS}\emptyset\text{L}$ has been called, the values obtained for the perturbations are retrieved from the column $\text{AA}(\text{I},\text{ISSP3})$. Then the $\delta \chi_i$ values, denoted by $\text{SDCHI}(\text{I})$, are calculated from equation I(360).

3.39 Function $\text{PI}\emptyset\text{MEG}$

This routine provides the Maxwell-averaged momentum transfer cross section for collisions of electrons with

neutral species as a function of the electron temperature, TE. The cross section is calculated by linear interpolation in a table stored in common block /TNEQ/.

3.40. Subroutine PRØP

The function of this routine is to compute the flow conditions at a specified temperature in the frozen flow solution. Since PRØP makes use of species thermal properties, a call to PRØP must always be preceded by a call to THERM.

The technique used by PRØP has been discussed in Section 6.4 of Volume I. The coding is straightforward. As noted in Section 3.22.1, the flow variables are computed in nondimensional form. In particular, equation I(267) for the pressure is written in the form

$$ZP = \ln \left(\frac{P}{P_0} \right) = -\ln P_0 + \sum_{j=1}^n X_{j0} \frac{S_{j0}}{R_0} - \sum_{j=1}^n X_{j0} \ln X_{j0} - \frac{W_0 S_0}{R_0} \quad (123)$$

The first sum on the right is evaluated in PRØP as

$$\sum_{j=1}^n X_{j0} \frac{S_{j0}}{R_0} = W_0 \sum_{j=1}^n \gamma_{j0} S_{j0} = CMA \cdot S4 \quad (124)$$

The remaining three terms on the right in (123) are computed as $W_0 \cdot ENT$, where ENT has been calculated in FRØZEQ before PRØP is first called:

$$ENT = - \sum_{j=1}^n \gamma_{j0} (\ln \gamma_{j0} + \ln W_0) - \frac{\ln P_0}{W_0} - \frac{S_0}{R_0} \quad (125)$$

3.41 Subroutine PRTA

This subroutine controls the output of the flow conditions and executes the calls to the MØDEL routine during

the nonequilibrium solution. It also calls subroutine BLCALL with the argument FINAL = .TRUE., to update the derivative of the boundary layer displacement thickness and provide permanent increments to the Cohen-Reshotko boundary layer integral. PRTA calls BLCALL even if the boundary layer is being neglected in the flow solution (ISW3B = 0), because the output and model routines need the species mole fractions, which are computed and loaded into common block /TEMPRY/ by BLCALL. For ISW3B = 0, the call to the boundary layer routine BLAYER is suppressed in BLCALL.

The step size in the nonequilibrium integration is often quite small, especially in the region upstream of the throat, where some of the reactions may still be nearly in equilibrium. To avoid the excessive output which would result if the flow conditions were printed out for every step in such regions of small step size, NATA prints the results of the nonequilibrium solution at intervals of TPRINT in the non-dimensional temperature CT. The control parameter TPRINT is preset to 0.01, and is under input control (with the input name TPRNTI). A counter (NSTEPS) is used to determine the number of steps computed between successive printouts of the flow conditions. This information is included in the printed output.

If the flow solution is being generated by the perturbation technique (INEQ = 0), the temperature perturbation PCT is added to CT before subroutine ØUT2 is called to produce the output. Then PCT is subtracted to restore CT to its equilibrium flow value.

3.42 Subroutine PUTQIN

The computation of the cross sections $\bar{\Omega}_{ij}^{(\ell,s)}$ required in the transport property calculations for any particular conditions of gas temperature, pressure, and composition is carried out in the NATA code under the general control of subroutine PUTQIN. The subroutine selects data for each step of the computations from the edited cross section data in common block /TRANS7/, calls the appropriate subroutine to carry out the computations, and then returns the computed

(1) cross sections for all species pairs to subroutine TRANSP in the Q(K,I,J) array of common block /TRANS1/ (see the discussion of subroutine TRANSP). The argument X of subroutine PUTQIN is the partial pressure of electrons in the specified gas mixture, in atmospheres, a quantity which is needed in the computation of effective Coulomb cross sections from equations I(100).

Before beginning the cross section computations, subroutine PUTQIN initializes Q(K,I,J) to zero for all pairs of species included in the cross section computations.* If the first entry in the V array is positive, as it should always be for proper operation of the code, Q(1,2,1) is set to $0.8 Q_C$, where Q_C is the quantity defined by equation I(100b), for later use in the computation of effective Coulomb cross sections for the mixture. The remainder of the subroutine down to statement 240 then consists of a loop which goes through the steps L of the cross section computation one by one and accumulates the values of the cross sections for all species included in the computations in the array Q(K,I,J). In this computation, the index $K = KQ(L)$ indicates the option to be used in performing the Lth step of the computations (see Section 4.6 of Volume II), M is the location of the first parameter for the Lth step of the computations in the parameter array V, and LQ1 and LQ2 are the locations respectively of the first and last pairs in the IQ, JQ array to which the computations of the Lth step are to be applied. For each step L of the computations, subroutine PUTQIN determines the option which is to be used for that step from the KQ array and then calls the appropriate subroutine to carry out the computations for that option, as indicated in Table II. For the Coulomb cross sections ($KQ = 2$) however, these computations are carried out in subroutine PUTQIN itself using the previously computed value $0.8Q_C$.

*Note that the number N of species in the cross section computations, which is stored in common block /TRANS8/, is not necessarily the same as the number of species in the transport calculations which is stored in common block /TRANS5/.

TABLE II

SUBROUTINES USED IN CROSS SECTION COMPUTATIONS

Option KKQ	Subroutine Used	Type
2	PUTQIN	1
3	QEXP	1
4	QEX	3
5	QTAP	1
6	QREPP	1
8	QLJ	1
9	QSAME	1
10	QMIX	2
11	Q11	2
12	Q12	1
13	Q13	2
14	Q14	2

As indicated in Table II, the subroutines called by PUTQIN may be divided into three general types, based on the manner in which they return their output. The first and most commonly used type, 1, returns the three cross section values $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$ and $B^* \bar{\Omega}(1,1)$ computed for the step to subroutine PUTQIN in the three locations $\emptyset M(1)$, $\emptyset M(2)$, and $\emptyset M(3)$ respectively of the dimensioned variable $\emptyset M$. The loop from statement 190 through 200 of PUTQIN then adds these computed values to the previously computed cross sections $Q(K, IQ, JQ)$, for each pair of species in the IQ, JQ array to which the step applies. For the type 2 subroutines, on the other hand, the cross sections are loaded directly into the Q array by the subroutine which calculates them, and statements 190 through 200 in PUTQIN are bypassed. The reason for this difference is that for the type 2 subroutines the computed values of the cross sections for a species depend on the indices of the species, so that a single set of cross section values cannot be returned for all pairs of species to which the step applies.

The single type 3 subroutine in the code, QEX (charge exchange cross section), returns the computed cross section values for the step in the variable ϕM as do the type 1 subroutines. However, in this case, the coding of subroutine PUTQIN offers a choice, determined by the input values of V for the option, .s to whether the computed cross sections for the step will be added to the previous cross sections by transferring control to statement 190, as for a regular type 1 subroutine, or will be used to replace the previous values of $\bar{\Omega}^{(1,1)}$ and $B^* \bar{\Omega}^{(1,1)}$ by transferring control to statement 170 and bypassing 190.

The final section of coding between statements 210 and 240 of PUTQIN produces a voluminous dump of cross section data under control of the input parameter ISW8B. For $ISW8B \leq 0$, this dump is not produced. For $ISW8B > 0$, the dump is produced the first time PUTQIN is called and every ISW8Bth time thereafter. Note that PUTQIN is called once each time subroutine TRANSP is executed to compute transport properties. For each step in the cross section calculations, the dump includes a line of output giving the number of the step; the code number KKQ of the cross section option used in the step; the positions in the IQ, JQ arrays of the first (IQ1) and the last (NQ(L)) pairs of species for which cross sections are calculated in the step; and the position M of the first unused parameter in the V array. Below this line of indexing information, the calculated values of the averaged cross sections $\bar{\Omega}_{ij}^{(1,1)}$ for all pairs of species in the mixture are printed out as they exist at the conclusion of the step. Together with the initial edit of the cross section calculations (Section 3.9 of Volume II), this PUTQIN dump allows detailed checking of the operation of the cross section calculations for a new gas model, since the effect of each step in the calculations upon each pair cross section $\bar{\Omega}_{ij}^{(1,1)}$ is clearly shown. The routine could easily be modified to print $\bar{\Omega}_{ij}^{(2,2)}$ and $B_{ij}^* \bar{\Omega}_{ij}^{(1,1)}$, in addition.

3.43 Subroutine QCUL

Subroutine QCUL computes the factor $0.8Q_c$ in the Coulomb cross section from equation I(100b). Inputs for this computation are the temperature T in $^{\circ}K$ and the

electron pressure $X = n_e kT$ in atmospheres. The computed value of $0.8Q_c$ in 10^{-16} cm^2 is returned in ØM1 . For $C \leq 0$, $\ln(f \Lambda)$ in $I(100b)$ is set to 1.0 while for $C > 0$ it is calculated correctly from $I(100c)$ and $I(100d)$. To prevent overflow when $n_e = 0$, a small quantity ($\approx 10^{-50} T^4$) is added to X before carrying out the computations.

3.44 Subroutine QEX

Subroutine QEX computes the averaged cross sections $\bar{\Omega}(1,1)$ and $B^* \bar{\Omega}(1,1)$ in 10^{-16} cm^2 for a pair of species between which there is an exchange interaction (i.e., an interaction such as charge exchange in which the identities of the two species are interchanged) having a cross section of the form

$$Q_{\text{ex}} = (A - B \log_{10} v)^2, \quad (126)$$

where v is the relative velocity in cm/sec . The formula for the averaged cross sections $\bar{\Omega}(l,s)$ is then (ref. 9):

$$\bar{\Omega}(l,s) = 2 [A - \alpha_s B - \frac{1}{2} B \ln(T/W)]^2 + 2 \beta_s B^2 \quad (127a)$$

where

$$\alpha_s = \frac{1}{2} [\ln(4k/N_0) + \sum_{n=1}^{s+1} \left(\frac{1}{n}\right) - \gamma] \log_{10} e$$

$$\beta_s = \frac{1}{4} \left[\frac{\pi^2}{6} - \sum_{n=1}^{s+1} \left(\frac{1}{n^2}\right) \right] (\log_{10} e)^2 \quad (127b)$$

$\gamma = 0.577216 \dots$ is Euler's constant, and W is the molecular weight. Putting the numerical values into equations (127) and $I(91)$ then gives the formulas used for $\bar{\Omega}(1,1)$ and $B^* \bar{\Omega}(1,1)$ in the subroutine. Since the cross section $\bar{\Omega}(2,2)$ is not affected by exchange (ref. 10), subroutine QEX sets the contribution $\text{ØM}(2)$ to this cross section equal to zero.

3.45 Subroutine QEXP

Subroutine QEXP computes the averaged collision cross sections $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$, and $B^* \bar{\Omega}(1,1)$ in 10^{-16} cm^2 for a pair of species interacting according to the exponential potential $\phi = Ae^{-r/\rho}$, where r is the distance between the two species in \AA . These cross sections are obtained by linear interpolation in Monchick's tables (ref. 11) of α versus $I(1,1)$, A^* , and B^* , where

$$\alpha = \ln (A/kT)$$

$$I(1,1) = \bar{\Omega}(1,1) / (4 \pi \alpha^2 \rho^2) \quad (128)$$

and

$$A^* = \bar{\Omega}(2,2) / \bar{\Omega}(1,1),$$

which are stored in common block /TRANS4/. The table lookup and interpolation are carried out by a call to subroutine QINTRP. The coding assumes that there are 50 entries in these tables.

3.46 Subroutine QINTRP

Subroutine QINTRP is used by several of the other cross section subroutines for interpolating tabular data stored in common block /TRANS4/. In this interpolation, the independent variable is assumed to be stored in monotonically increasing order in the array TL, starting at location N1, and the corresponding values of the three dependent variables in the corresponding locations of the OMEGA1, ASTAR and BSTAR arrays. The argument N of the subroutine represents the number of data points to be included in the table, and A and B are respectively the values of the independent and dependent variables for the interpolation. The subroutine searches the data in the TL array for the specified value A of the independent variable and, if A is found to lie within the range of the table, carries out a linear interpolation of the tabular data to determine the values B(J) of the three dependent variables corresponding to the specified value A of the independent variable. If, on the other

hand, A lies outside the range of the table, the last two data points in the table are linearly extrapolated to determine the values of the dependent variable B and, if ISW8B \neq 0, a message is printed out indicating that an extrapolation of the data was required. This message gives the location in the TL array at which the extrapolation occurred, the desired value A of the independent variable, and the two values of the variable from the TL array used in the extrapolation.

3.47 Subroutine QLJ

Subroutine QLJ determines the collision cross sections $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$ and $B^* \bar{\Omega}(1,1)$ at temperature T for a pair of species interacting according to the Lennard-Jones 6-12 potential $\phi = 4 \epsilon [(\sigma/r)^{12} - (\sigma/r)^6]$ by linear interpolation in tables of $T^* = kT/\epsilon$ versus $\bar{\Omega}(1,1)^* = \bar{\Omega}(1,1)/\pi\sigma^2$, $A^* = \bar{\Omega}(2,2)/\bar{\Omega}(1,1)$, and B^* which have been computed for this potential by Monchick and Mason (ref. 12). These tables contain data for 37 reduced temperatures T^* from 0.1 to 100 and are stored in the arrays TL, OMEGA1, ASTAR, and BSTAR respectively in common block/TRANS4/, starting at location N1 = 501 in the arrays.

3.48 Subroutine QMIX

Subroutine QMIX computes cross sections from the empirical mixing rule I(102) and adds them to the previously computed cross sections Q(K,I,J) for a series of species pairs given in the IQ, JQ arrays in common block/TRANS7/. The arguments LQ1 and LQ2 of the subroutine indicate the positions of the first and last pairs in the IQ, JQ array to which these computations are to be applied.

To avoid computing the square roots required in equation I(102) several times in a single call to QMIX, these quantities are stored in the array

$$SQT(K,I) \equiv \frac{1}{2} \sqrt{Q(K, I, I)} \quad (129)$$

Upon each entrance to the subroutine, the values of $SQT(1,I)$ are set to 0.0 for all species I , to indicate that they have not yet been calculated in the current call to QMIX. (Note that values of SQT cannot be saved over from one call to QMIX to the next, since the values of the cross sections $Q(K,I,I)$ may have changed between the two calls.) The remainder of the subroutine then consists of a loop over the species pairs I, J which are to be included in the computation. For each pair, the subroutine checks the values of $SQT(1,I)$ and $SQT(1,J)$ to determine if they have been calculated previously in the loop, and, if not, computes the required values of SQT from (129) for $K = 1, 2$, and 3. The SQT values for the pair are then used in equation I(102) and the results added to the previously computed cross sections $Q(K,I,J)$ to obtain the final cross section values for the pair.

Since the edited cross section data prepared by subroutine XSECT may contain steps using the option $KQ = 10$ which do not apply to any species pairs, a provision to bypass the cross section computations completely when $IQ1 > IQ2$ is included in the subroutine QMIX to allow for this possibility

3.49 Subroutine QREPP

Subroutine QREPP computes the averaged cross sections $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$, and $B * \bar{\Omega}(1,1)$ in units of 10^{-16} cm^2 for a pair of species which interact according to an attractive or repulsive inverse power potential of the form $\phi = Ar^{-\eta}$, where r is the distance between the species in \AA . The cross section formulas for this potential are (ref. 13):

$$\begin{aligned}\bar{\Omega}(1,1) &= \phi \text{MEGA1(ITL)} / (T^{2/\eta}) \\ \bar{\Omega}(2,2) &= \text{ASTAR(ITL)} * \bar{\Omega}(1,1) \\ B * \bar{\Omega}(1,1) &= \text{BSTAR(ITL)} * \bar{\Omega}(1,1)\end{aligned}\tag{130}$$

where $\phi \text{MEGA1(ITL)}$, ASTAR(ITL) , and BSTAR(ITL) are input quantities given by equations I(105).

3.50 Subroutine QSAME

Subroutine QSAME sets the cross sections for a species pair equal to a constant multiple of those computed previously for another pair (I,J). The coding of the subroutine permits a different constant factor to be used for each of the three cross sections $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$, and $B^* \bar{\Omega}(1,1)$.

3.51 Subroutine QTAB

Subroutine QTAB obtains the values of the cross sections $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$, and $B^* \bar{\Omega}(1,1)$ for a pair of species at a given temperature T by linear interpolation in tables of $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$ and B^* versus temperature for the species which are included in the input. The tabulated data are assumed to be given in the arrays OMEGA1, ASTAR, BSTAR, and TL, respectively, in common block/TRANS4/, with the tables beginning at location N1 in the arrays and containing NL data points. An option is also provided to multiply the tabulated cross sections by an arbitrary constant factor A in the computations.

3.52 Subroutine Q11

Subroutine Q11 multiplies the previously computed cross sections for a series of species pairs by the temperature dependent factor $f(T)$ given by equation I(108). The subroutine computes the value of this factor at the specified temperature T from the input data for the subroutine and then calls subroutine Q14 with the appropriate inputs to carry out the actual multiplication of the cross sections by this factor.

3.53 Subroutine Q12

Subroutine Q12 computes the cross sections $\bar{\Omega}(1,1)$, $\bar{\Omega}(2,2)$, and $B^* \bar{\Omega}(1,1)$ for a species pair from the generalized mixing rule given by equation I(109). As with the other type 1 subroutines discussed above, it returns the computed cross sections to subroutine PUTQIN in the dimensioned variable OM.

3.54 Subroutine Q13

Subroutine Q13 computes the cross sections $Q(K1, I, J)$ according to equations I(110) and I(111) for a fixed value of $K1$ specified in the cross section data ($1 \leq K1 \leq 3$) and for a series of species pairs I, J given in positions LQ1 through LQ2 of the IQ, JQ arrays. The computed cross sections then replace the previous values for these cross sections in the $Q(K, I, J)$ array.

3.55 Subroutine Q14

Subroutine Q14 computes the cross sections $\bar{Q}(1,1)$, $\bar{Q}(2,2)$, and $B^* \bar{Q}(1,1)$ for a series of species pairs by multiplying the previously computed values of these cross sections by a constant factor, as indicated in equation I(112). The computed cross sections then replace the previous values for these cross sections in the $Q(K, I, J)$ array.

3.56 Subroutine RADIUS

This small routine performs geometric calculations that are required at several points in the computation of the boundary layer at the first flow point in subroutine BLAYER. The inputs to RADIUS are the type of nozzle geometry ITYPE, the axial coordinate X, and the profile index L. The quantities computed and returned to BLAYER are the profile ordinate R, the geometric area ratio AG, and the factor AGJ, which is proportional to r^2_j in equation I(171).

For ITYPE = 1 (two-dimensional nozzle), R is obtained by calling the entry GMAR of subroutine GEOMAR. Then $AG = R/R_0$ where R_0 is the half-width of the throat gap, and $AGJ = 1$ since $j = 0$.

For ITYPE = 2 (axisymmetric nozzle), R is again obtained by calling GMAR, $AG = (R/R_0)^2$, where R_0 here is the throat radius, and $AGJ = AG$ since $j = 1$.

For ITYPE = 3 (rectangular channel), the ordinates $YZ(1)$, $YZ(2)$ of the two profiles are obtained by calling the entry GMAR2 of subroutine GEOMAR, $AG = YZ(1) \cdot YZ(2) / Y_0 Z_0$ where $Y_0 Z_0$ is the product of the profile ordinates at the throat, $R = YZ(L)$, and $AGJ = [YZ(M)]^2$, where $M = 2$ if $L = 1$ and $M = 1$ if $L = 2$.

3.57 Subroutine READ

READ is the input routine for NATA. It reads the input data, sets up the geometric description and gas model for the problem, and initializes most of the control parameters.

3.57.1 Input Operations

The subroutine begins by reading the input data for the current case. The routine contains four READ statements, of which the first two are always executed. The first of these statements reads a comment card into the array ACØM. This information is subsequently written out as part of a heading for the output for the case. The second READ statement reads data for the variables included in namelist INPUT. This is the main input operation for the code. Namelist INPUT contains all of the inputs to NATA except those required for defining chemical elements, species, and reactions (read under namelist EINPUT), and those required for calculating the transport cross sections of species (read under namelist TINPUT). When the code is run using the chemical and cross section data compiled into the BLOCK DATA routines, as it is normally, the statements READ(5, EINPUT) and READ(5, TINPUT) are not executed.

All of the variables in INPUT are printed out at the head of the output for the case by the statement WRITE (6, INPUT). If EINPUT and TINPUT are read, their variables are also printed out in namelist format. The reason for breaking the code inputs into these three groups is to minimize the number of pages of such output in normal usage, when EINPUT and TINPUT are not used.

3.57.2 Sonic Mass Flux Calculation

If $ISW2B \leq 0$, NATA determines the equilibrium conditions in the reservoir from data on the reservoir pressure or stagnation enthalpy and the total mass flow. In these cases, the sonic mass flux SMASS is required in the calculations to determine the reservoir temperature. This quantity is computed in READ from the total mass flow \dot{M} (denoted by FLØW) and the geometric dimensions of the throat. The coding treats three cases:

(1) Two-dimensional nozzle

Here \dot{M} and FLØW denote the mass flow per unit length of the throat gap. Hence

$$SMASS = \frac{\dot{M}}{D} \quad (131)$$

where D is the throat gap. Since SMASS is required in gm/cm²-sec while \dot{M} is in lb/in-sec and D in inches, a numerical conversion factor from lb/in²-sec to gm/cm²-sec is inserted on the right. This factor is $453.5924/(2.54)^2 = 70.3069$.

(2) Axisymmetric nozzle

In this case, \dot{M} is the total mass flow in lb/sec, and

$$SMASS = \frac{4\dot{M}}{\pi D^2} \quad (132)$$

Insertion of the conversion factor (70.3069) gives a numerical coefficient of $4 \times 70.3069/\pi = 89.5173$.

(3) Rectangular channel

The sonic mass flux here is

$$SMASS = \frac{\dot{M}}{D_1 D_2} \quad (133)$$

where D_1 and D_2 are the throat gaps of the channel's two profiles.

3.57.3 Automatic Air Model Selection

If the input AAMS (acronym for "automatic air model selection") is .TRUE., and IGAS is either 1 or 2, subroutine READ resets IGAS to either 1 or 2 on the basis of an enthalpy or temperature criterion. IGAS = 1 gives the large air model, including five ion species, which is required in cases

with high reservoir temperatures. IGAS = 2 gives the small air model which includes only a single ion (NO^+) and which can be used at lower temperatures. If the reservoir temperature is read in (ISW2B > 0), the air model is selected to be IGAS = 1 or 2 according as the reservoir temperature CTAPI is greater than or less than 6000°K . On the other hand, if the reservoir temperature is to be determined from mass flow data (ISW2B \leq 0), then its value is not yet known at the time when READ is executed. In this case, the air model selection is based on a stagnation enthalpy criterion of 8000 Btu/lb. For ISW2B < 0, the stagnation enthalpy is an input. For ISW2B = 0, the inputs are stagnation pressure and mass flow; in this case, the stagnation enthalpy is estimated from the Winovich correlation (ref. 14). The Winovich estimate of the stagnation enthalpy is computed from the formula

$$H_{\text{WINOV}} = \left[\frac{136.7 p_0}{m_*} \right]^{2.52} \quad (134)$$

where p_0 denotes the reservoir pressure in atmospheres and m_* the sonic mass flux in $\text{g/cm}^2 \text{ sec}$. The numerical factor on the right in (134) differs from the coefficient (280) in Winovich's equation because it includes a factor converting m_* from $\text{g/cm}^2\text{-sec}$ to $\text{lb/ft}^2\text{-sec}$.

3.57.4 Initializations Based on Inputs

With few exceptions, the input variables of NATA are not altered by execution of the program. Also, the Namelist input system resets only those variables which are actually referenced in the input cards for a given case. Thus, when several cases are run in a single job, the input cards for the second and subsequent cases need contain data only for the variables which the user wishes to change, e.g., the pressure, mass flow, nozzle index, etc.

The input-determined variables which are not altered by program execution are included directly in the namelists INPUT, EINPUT, and TINPUT. However, some input-determined variables are altered by the program. In such cases, the variable which is altered and its corresponding input are given different names and thus are stored in separate

locations. The input name is the internal variable name (or a contraction thereof) with the letter I added at the end. At the cost of some increase in storage requirements, this arrangement protects the input variables against alteration during execution.

3.57.5 List of Test Sections

The data in the input array TSDIAM specify points in the flow at which special calculations are to be done. In the case of a two-dimensional or axisymmetric nozzle, the entries in TSDIAM are nozzle diameters at which model-condition calculations are to be performed. For a channel, the TSDIAM entries are specified channel widths at which free-stream flow and boundary layer calculations are to be performed.

Subroutine READ uses the data in the TSDIAM array to set up another array, TSAR, whose values are actually used by the program to select the points at which the special calculations are done. For a nozzle, the TSAR entries are geometric area ratio values. For a channel, the TSAR are widths in centimeters. In generating the TSAR array, READ rejects any TSDIAM values which are less than or equal to the throat diameter for the profile involved, and determines the number (NTS) of valid TSAR entries. All the entries of TSDIAM are preset (in BLOCK DATA) to the value 1×10^{20} . Thus, any values in excess of 1×10^{20} are assumed not to have been set by the user in input, and are not regarded as valid points.

3.57.6 Geometric Description

A nozzle is specified by one profile curvefit, a channel by two. For each profile, the parameters defining the geometry of the nozzle or channel may be either read in as part of the input or, if one of the standard geometries is desired, obtained from precoded data. For the Jth profile ($J = 1$ or 2), if $\text{NPRØFL}(J)$ is nonzero, the geometric data are looked up in the ZPRP array, which is set in BLOCK DATA. This array occupies the common block /NØZZ/. If $\text{NPRØFL}(J)$ is zero, the geometric data are set on the basis of the geometric input variables such as NSECTS, DIAM, and PARAMI.

In either case, after the geometric model has been set up, subroutine READ prints out a summary description of it, unless the standard geometry specified is the same as in the preceding case in the job.

3.57.7 Extra Chemical Elements

If additional chemical elements have been defined in the EINPUT input, the data for them are loaded into the EPRP array, which contains the precoded data for elements. The number of input-defined elements is NEELS. For the Ith such element, the atomic number is obtained from EEP RP(1,I) and the atomic weight from EEP RP(2,I). The index assigned to the element in the master list of elements is IEEP(I). The Hollerith atomic symbol for the element is obtained from the array ASYM, which is set in a data statement in READ.

3.57.8 Gas Model

The elements, species, and reactions present in the gas mixture are specified by the group of variables ISC, ISS, ISR, IC, IE(I), NSC, QPJ(I), JCS(I), IS(I), and IR(I). If IGAS=0, these variables are obtained from the input data. If IGAS is positive, they are looked up in the array of precoded data, GPRP(I,IGAS), which occupies common block /MIXT/. The physical definitions of the entries in this array are given in Section 4.5 of Volume II. If IGAS is negative, the input value is saved in the location IGS, and IGAS is reset to its own absolute value. In this case, all of the above-listed variables except QPJ(I) are again looked up in GPRP, but the QPJ (the mole fractions of the cold species) are obtained from input. This provision allows input control of the overall elemental composition of the standard gas mixtures containing more than one chemical element.

After these general specifications of the gas model have been determined, the required properties of the chemical elements, species, and reactions are looked up. The data for elements are obtained from the array EPRP(I,IE), where the second subscript is the index in the master list of elements. The species data are looked up in the array SPRP(I,IS), where IS is the index in the master list of species. The reaction data are obtained from RPRP(I,IR),

where IR is the index in the master list of reactions. The entries in these arrays are all defined in Section 4 of Volume II. The order of the elements, species, and reactions in a gas model is determined by the index arrays IE(I), IS(I), IR(I), and may differ from the order in the corresponding master lists.

3.57.9 Geometric Sequence of Model Points

Positions at which model condition calculations are to be done can be determined by specifying an initial point XMØDPl, a final point CXMAXI, and a number of points NMØDPT. From these data, subroutine READ computes a factor FACMP which is used to generate a geometric sequence of NMØDPT values of x extending from XMØDPl to CXMAXI. This factor is calculated from the formula

$$\text{FACMP} = \left[\frac{\text{CXMAXI}}{\text{XMØDPl}} \right]^{1/(\text{NMØDPT}-1)} \quad (135)$$

3.58 Subroutine RESET

This Fortran routine, used only in the IBM version of NATA, simulates the UNIVAC 1108 timing routines RESET and TIME. The time data are obtained by calling the library routine ACUCPU, which returns in its argument ICPU the number of milliseconds of CPU time remaining before the TIME parameter of the job step is reached. Subroutine RESET is called by the main program near the beginning of each NATA job. The initial value of ICPU is stored in IZERØ. Subsequent calls to the entry TIME then give the elapsed time I in milliseconds.

3.59 Subroutine RESTMP

RESTMP performs the iterative solution for the reservoir conditions when the second or third option for input specification of these conditions is used. These options have been described in Section 6.5 of Volume I. Usage of these options is controlled by the indicator ISW2B. For ISW2B = 0, the second option is used, and for ISW2B < 0, the third option. (For ISW2B > 0, the first option is used and RESTMP is not called.)

3.59.1 Second Option

The second option is implemented by the coding down to statement 90. In this option, the input specification of the reservoir conditions is based on the reservoir pressure, p_0 , and the total mass flow, \dot{M} . In the code, the value of p_0 in atmospheres is available in the location PRESA in unlabelled common, and the sonic mass flux m_* in $\text{g/cm}^2\text{sec}$ is given by SMASS in common block /MASSFL/.

The basic problem in the second option is to determine the reservoir temperature, T_0 , from these data. The code capabilities available to support this effort are

- (1) Entry INTA of subroutine INGAS, which computes all of the conditions in the reservoir from PRESA = p_0 and CTAP = T_0 .
- (2) Subroutine NRMAX, which computes the sonic mass flux m_* for equilibrium flow, based on given reservoir conditions.

RESTMP determines T_0 by an iteration in which INTA and NRMAX are called, at each step, and T_0 is repeatedly adjusted to make the calculated sonic mass flux nearly equal to SMASS.

RESTMP starts the iteration with a standard guess $T_0 = 10000^\circ\text{K}$. Subroutines INTA and NRMAX are then called to compute the reservoir conditions and sonic mass flux based on the trial value of T_0 . The call to INTA must be preceded by a call to subroutine INIT to set a number of species properties which are nondimensionalized using the reservoir temperature.

The sonic mass flux computed by NRMAX is in the non-dimensional form

$$SM = \frac{\rho_* u_*}{\rho_0 u_s} = \frac{m_*}{\rho_0 \sqrt{R_0 T_0} / W_0} \quad (136)$$

From the perfect gas law,

$$p_0 = \frac{\rho_0 R_0 T_0}{W_0} \quad (137)$$

Elimination of ρ_0 between (136) and (137) gives the relation

$$T_0 = \frac{W_0(T_0)}{R_0} \left[\frac{p_c \cdot SM(T_0)}{m_*} \right]^2 \quad (138)$$

which is satisfied by the correct value of the reservoir temperature T_0 . In (138), W_0 denotes the reservoir molecular weight (CMA), R_0 the universal gas constant in mechanical units (8.31434×10^7 erg/mole $^\circ K$), p_c the reservoir pressure in dyne/cm², $SM(T_0)$ the nondimensional sonic mass flux computed by NRMAX, and m_* the sonic mass flux (SMASS, g/cm²sec) based on the input total mass flow \dot{M} . Equation (138) may be rewritten

$$T_0 = \frac{W_0(T_0)}{C} [SM(T_0)] = \frac{F(T_0)}{C} \quad (139)$$

where the parameter C (denoted by CONST in the program) is a constant of the iteration:

$$C = R_0 \left[\frac{m_*}{p_c} \right]^2 \quad (140)$$

The second estimate of T_0 is obtained using equation (139), with W_0 and SM values based on the first trial value. The quantity SM is found to vary only weakly with T_0 , while the molecular weight is monotonically decreasing function of T_0 at constant pressure. Thus, the error in the second estimate of T_0 is opposite in sign to that of the first estimate; the two values bracket the correct solution. Accordingly, after INIT, INTA, and NRMAX have been called, the third estimate of T_0 is obtained by a linear interpolation technique. The function $F(T_0)$ is approximated as a linear function between $T_0 = T_1$ and $T_0 = T_2$ (the first two estimates). Then

$$\frac{T_0 - T_2}{T_0 - T_1} = \frac{F_2 - F_1}{T_2 - T_1} \quad (141)$$

Elimination of F between (139) and (141) gives

$$T_0 = - \frac{T_2 F_1 - T_1 F_2}{T_2 - T_1} \cdot \frac{1}{\frac{F_2 - F_1}{T_2 - T_1} - C} \quad (142)$$

for the third estimate.

Beginning with the fourth estimate, the dependence of $F(T_0)$ upon T_0 is approximated by a quadratic function based upon the data from the three most recent iterations:

$$F = D_1 + D_2 T_0 + D_3 T_0^2 \quad (143)$$

Substitution of the data from the three points gives a system of linear equations for D_1, D_2, D_3 . This system can be written in matrix form

$$A \cdot D = B \quad (144)$$

where

$$A = \begin{bmatrix} 1. & T_1 & T_1^2 \\ 1. & T_2 & T_2^2 \\ 1. & T_3 & T_3^2 \end{bmatrix} \quad (145a)$$

$$B = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}, \quad D = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} \quad (145b)$$

This system of equations is solved using subroutine SIMQ, which places the solution vector D into the locations originally occupied by the vector of constants B . Then elimination of F between (139) and (143) gives

$$T_0 = C (D_1 + D_2 T_0 + D_3 T_0^2) \quad (146)$$

The solution of this quadratic equation for T_0 is

$$T_0 = \frac{1}{2D_3} \left[- (D_2 - \frac{1}{C}) \pm \sqrt{(D_2 - \frac{1}{C})^2 - 4 D_1 D_3} \right] \quad (147)$$

The sign before the radical is determined by noting that, for $D_3 \rightarrow 0$, the solution must approach

$$T_0 = - \frac{D_1}{D_2 - 1/C} \quad (D_3 = 0) \quad (148)$$

Thus, the sign represented by \pm must be opposite to that of $-(D_2 - 1/C)$, i.e., the same as that of $(D_2 - 1/C)$.

In the program, the temperature values T_1, T_2, T_3 are stored in the array CTSAVE and the parameter values F_1, F_2, F_3 in the array F. If SIMQ returns an error indicator $KS \neq 0$, which will occur if the matrix (145a) is singular, RESTMP reverts to the linear interpolation technique. The convergence criterion for the iteration is that the relative change in T_0 between two successive iterative steps be smaller than or equal to 10^{-4} .

3.59.2 Third Option

The coding of the third option begins with statement 90 of RESTMP and continues to the end of the subroutine. In this option, the reservoir conditions are specified by input of the total mass flow \dot{M} and the stagnation enthalpy h_0 . The latter quantity is given, in calories per gram, by the variable HS in common /RDMAN/. The third option requires a double iteration to determine both the reservoir temperature T_0 and the reservoir pressure p_0 from the available data.

To start the iteration, an initial estimate of the reservoir pressure is obtained from an approximate sonic flow analysis for a perfect gas. For a gas with constant specific heats,

$$\rho_* = \left[\frac{2}{\gamma + 1} \right]^{1/(\gamma - 1)} \rho_0 \quad (149a)$$

$$u_* = a_* = \sqrt{\frac{2}{\gamma + 1}} a_0 \quad (149b)$$

$$a_0 = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\frac{\gamma R_0 T_0}{W_0}} = \sqrt{(\gamma - 1) h_0} \quad (149c)$$

These relations are derived, for example, in Liepmann and Roshko (ref. 15, pp. 51-54). Combination of equations (149) and (137) gives

$$p_0 = \frac{\gamma - 1}{\gamma} \sqrt{\frac{\gamma + 1}{2(\gamma - 1)}} \left[\frac{\gamma + 1}{2} \right]^{1/(\gamma - 1)} m_* \sqrt{h_0} \quad (150)$$

In RESTMP, an initial estimate of p_0 is obtained from equation (150) using a standard estimate of the specific heat ratio, $\gamma = 4/3$. This value gives

$$p_0 = 0.743 m_* \sqrt{h_0} \quad (151)$$

in which all quantities are assumed to be expressed in absolute cgs units. With p_0 in atmospheres and h_0 in cal/gm, this relation becomes

$$p_0(\text{atm}) = 4.8 \times 10^{-3} m_* \sqrt{h_0(\text{cal/g})} \quad (152)$$

which is the form used in RESTMP.

The iteration scheme used to solve for the reservoir temperature and pressure is illustrated schematically in figure 25. The starting point for the iteration (labelled "1") lies at the pressure (152) and an initial estimate of temperature, 10,000°K. In the first stage of the double iteration, the temperature is varied at constant pressure to make the reservoir enthalpy equal to the input value HS (denoted by h_g in the figure). The coding of this stage runs from statement 110 down to (but not including) 130. At each step of this iteration on temperature, INIT and INTA are called, and the reservoir enthalpy is computed from CHA using equation I(241). In the first step, the temperature

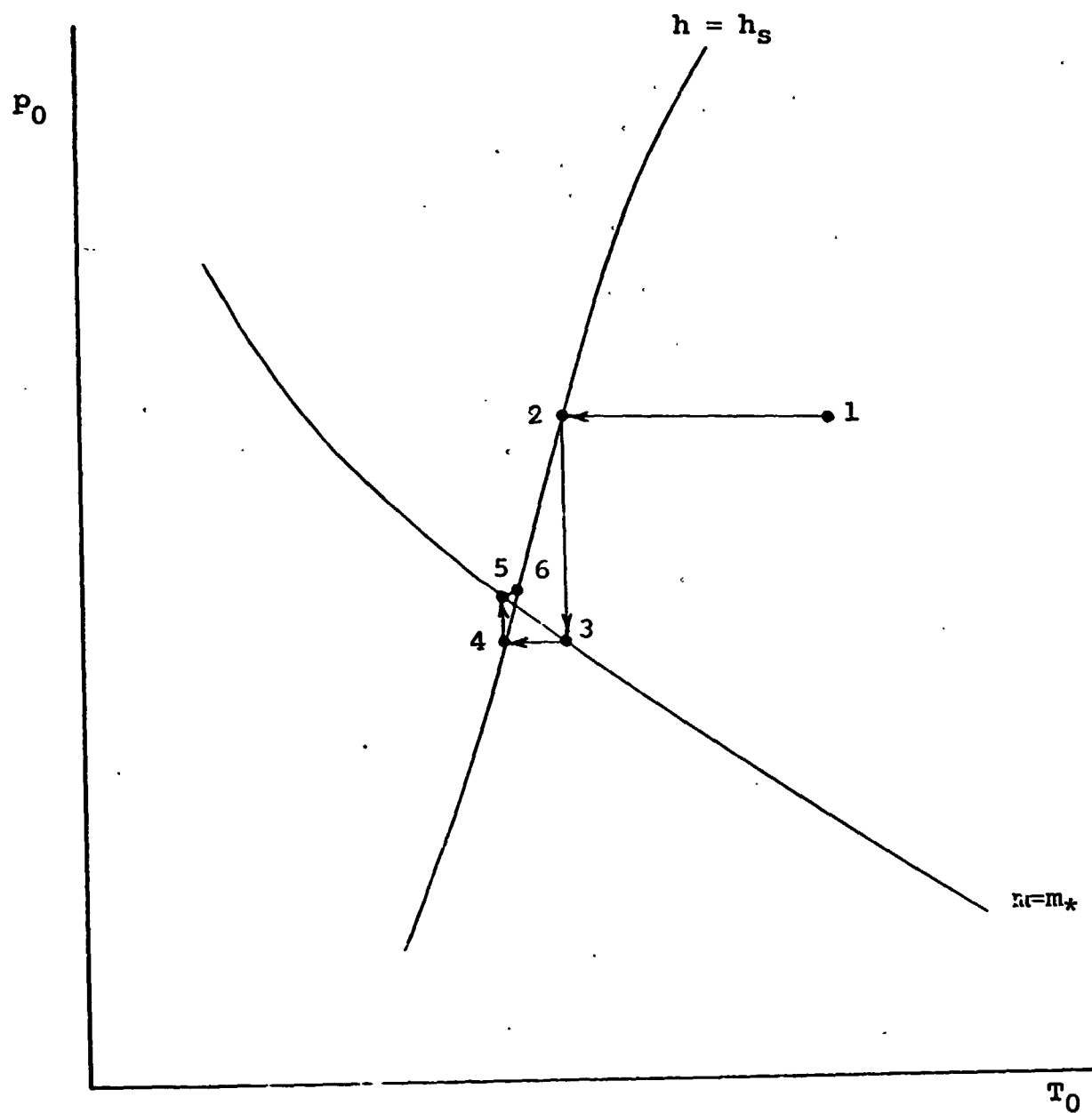


Figure 25. Iteration Scheme of Third Option for Specifying Reservoir Conditions

estimate is improved by assuming that the enthalpy is approximately proportional to the temperature:

$$T_0 = T_1 \cdot \frac{h_s}{h_1} \quad (153)$$

where T_1 is the initial estimate of T_0 and h_1 is the specific enthalpy corresponding to T_1 . In subsequent steps, the relation between T and H is approximated by a linear relation which does not necessarily pass through the origin:

$$T_0 = T + (h_s - h) (T_p - T) / (h_p - h) \quad (154)$$

This iteration on temperature is continued until the calculated enthalpy is equal to the desired value to within 1 part in 10^5 . The result of this stage is the point "2" in figure 25.

In the second stage, the reservoir pressure is varied at constant reservoir temperature to make the sonic mass flux equal to its desired value, SMASS. This stage is implemented by the coding from statement 130 down to (but not including) 150. In each cycle of this stage of the iteration, subroutine NRMAX is called to compute the non-dimensional sonic mass flux, SM. The sonic mass flux m^* in $\text{g/cm}^2\text{sec}$ is calculated from this value using equation (136). The estimate of the reservoir pressure is then improved on the assumption that m^* is directly proportional to p_0 , and subroutine INTA is called to determine the reservoir conditions based on the new p_0 value. This stage is continued until the calculated sonic mass flux agrees with the desired value to within 1 part in 10^5 . The resulting point is "3" in figure 25.

The complete iteration consists of alternate temperature-adjusting and pressure-adjusting stages, and produces a series of points "4", "5", "6", etc., which spiral in towards the desired solution point in $T_0 - p_0$ space (figure 25). After each pair of stages (2,3), (4,5), etc., tests on temperature and pressure are applied to assess the convergence of the entire procedure. The criteria for convergence are that the relative changes in T_0 and p_0 during a stage-pair be less than or equal to 10^{-4} .

If the input control parameter ISW A is nonzero, a dump is written in each cycle of this iteration to allow tracing of the subroutine's operation.

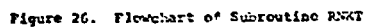
3.60 Subroutine RNKT

RNKT is the routine that computes the changes in the dependent flow variables over an integration step using the Treanor-Runge-Kutta scheme explained in Section 7.5.1 of Volume I. Figure 26 is a flowchart of RNKT.

In the initializations, IFAIL is an indicator for the cause of step failure, used in diagnostic output which is produced when ISW5B is set to a negative value in the input (see glossary of Fortran symbols for common block /TNERK/ in Section 4). TE is a logical indicator. A .TRUE. value means that a chemical nonequilibrium model is in use. If TE = .FALSE., an electronic equilibrium model is being used. LIM denotes the number of dependent variables included in the numerical integration. For a chemical nonequilibrium model, $LIM = n + 1$ and the variables are γ_j for $j = 1$ to n and $CT = T/T_0$. For an electronic nonequilibrium model, $LIM = n + 3$ and the variables are γ_j , CT , $CTE = T_e/T_0$, and $CHA = h_0 W_0 / R_0 T_0$. The energy transfer to the electron gas (QDPE) at the start of the integration step is denoted by QDPEB. Finally, DQMRK is the maximum allowable change in QDPE from QDPEB during the integration step.

For ISW5B different from zero, diagnostic dumps are produced at five locations in RNKT. The coding used to generate these dumps is omitted from the flowchart, as it is not essential to the function of the subroutine.

The computations in RNKT closely follow the analysis given in Section 7.5.1 of Volume I. The values y_1, y_2, y_3, y_4 are represented by arrays GJ1(J), GJ2(J), GJ3(J), GJ4(J), where J runs from 1 to LIM. Thus, these arrays contain not only the species concentrations but also the other dependent variables, CT, etc. The derivatives f_1, f_2, f_3 are represented by arrays F1(J), F2(J), F3(J). The parameter $(-P \Delta x)$ in I(398) is represented by an array P(J). The quantities F_1, F_2, F_3 defined by equations I(398) are represented by X1(J), X2(J), X3(J).



-175-

Following the initializations, RNKT loads data into the F1, GJ1, and GJ2 arrays. If any of the GJ2 values is negative, FAILED is set to .TRUE. and the RETURN is executed. If not, CX is set equal to CXB + HDELX, where HDELX is one-half of the step size DELTAX. Then DERIVS is called to compute the derivatives F2(J), and the GJ3 array is set. The signs of all the GJ3(J) values are tested. In addition, if TE is .FALSE., the new value of the energy transfer QDPE to the electron gas, produced by the call to DERIVS, is tested to determine whether $|QDPE - QDPEB|$ is less than or equal to DQMRK. If not, the error return is executed unless DQMAX = 0 (which means that the current step is the first step in the numerical integration).

If the solution thus far passes these validity checks, DERIVS is called again to determine the derivatives F3(J). Next, the parameters P(J) are calculated. The basic formula for P(J) is I(402). In the Fortran notation, with $P(J) \rightarrow -P \Delta x$ in I(402), this equation becomes

$$P(J) = 2 \cdot \frac{F3(J) - F2(J)}{F2(J) - F1(J)} \quad (155)$$

However, under certain conditions, P(J) is set to zero. As explained in Section 7.5.1 of Volume I, this is done if $(f_2 - f_1)/f_1 = X4(J)/F1(J)$ has a magnitude smaller than 10^{-4} , to avoid excessive loss of accuracy on subtractions. P(J) is also set to zero if F1(J) = 0, to avoid a divide check error in the test just described, or if the evaluation of (155) would lead to a floating point overflow on the indicated division. For $P(J) \geq 0$, the changes of the dependent variables are computed using the Runge-Kutta formula I(392) in place of Treanor's formula I(401).

Once the P(J) have been determined, RNKT proceeds to compute X1(J), X2(J), and X3(J). If $P(J) \leq -1.25$, these quantities are calculated from equations I(398). Since the Fortran P(J) corresponds to $(-P \Delta x)$ in equations I(398), these formulas can be rewritten in the form

$$F_1 \equiv X1(J) = \frac{e^{P(J)} - 1}{P(J)} \quad (156a)$$

$$F_2 \equiv X2(J) = \frac{X1(J) - 1}{P(J)} \quad (156b)$$

$$F_3 \equiv X3(J) = \frac{X2(J) - \frac{1}{2}}{P(J)} \quad (156c)$$

which is used in RNKT. If $P(J) > -1.25$ (but negative), the subtractions in (156) could cause excessive loss of accuracy. In this case, F_1 , F_2 , and F_3 are evaluated from the power-series expansion

$$F_n = \sum_{k=0}^{\infty} \frac{[P(J)]^k}{(n+k)!} \quad (157)$$

which can easily be derived by substituting the Maclaurin expansion for e^x into (156). In RNKT, the sums (157) for F_1 , F_2 , F_3 are accumulated in the locations S_1 , S_2 , S_3 . The general terms in the three expansions are denoted by T_1 , T_2 , and T_3 , respectively. These terms are initialized to their values for $k = 0$, namely $T_1 = 1$, $T_2 = 1/2$, $T_3 = 1/6$. Then the successive terms in the series are generated recursively:

$$(T_n)_k = (T_n)_{k-1} \cdot \frac{P(J)}{n+k} \quad (158)$$

where $(T_n)_k$ denotes the k^{th} term in the expansion for F_n . In RNKT, the denominator $(n+k)$ in (158) is represented by $(n+Z-3)$, where Z is initialized to 3 for $k = 0$ and is incremented by 1 each time around the iterative loop. The

evaluation of (157) proceeds until the addition of T3 to S3 no longer affects the value of S3 to within floating-point accuracy.

Next, the GJ4(J) array is set. For each J, if P(J) is negative, GJ4(J) is computed from equation I(405). For zero or positive P(J), GJ4(J) is calculated from equation I(394c). After the usual validity checks, the flow variables GJ(J), CT, etc., are set to their corresponding values in GJ4(J), CX is set to CXB + DELTAX, and a final call to DERIVS is executed. The resulting values for the derivatives of the flow variable are loaded into F4(J), and the array SDQ(J) of changes in the dependent variables over the whole integration step is computed. For each J, if $P(J) \geq 0$, the Runge-Kutta formula I(392) is used to calculate SDQ(J). If $P(J) < 0$, then Treanor's formula I(401) is used. Finally, the results are transferred from the SDQ(J) array into the storage locations SDGJ(J), SDT, etc., used by other parts of the program.

3.61 Subroutine SHOCK

Subroutine SHOCK* computes the oblique shock angle and the conditions behind the shock for supersonic flow of a perfect gas over a wedge. The first three of the arguments are inputs to the routine. BMF denotes the free-stream Mach number M, DEL the flow deflection angle (i.e., the angle of attack of the wedge surface, α), and GAM the specific heat ratio γ . The outputs are FL and IERR, which are defined in Section 4.128 and by comment cards in the subroutine listing.

The method of solution is based upon the relation between the shock angle σ and the deflection angle α , which is given, for example, in Liepmann and Roshko (ref. 15, p.87). This relation can be expressed as a cubic equation in $\sin^2 \sigma$:

*This subroutine was programmed by Harvey Buss, Avco Systems Division. Statements for calculating quantities not required in NATA have been converted into "comments" to save storage.

$$\sin^6 \sigma + b \sin^4 \sigma + c \sin^2 \sigma + d = 0 \quad (159)$$

where

$$b = -\frac{M^2+2}{M^2} - \gamma \sin^2 \alpha \quad (160a)$$

$$c = \frac{2M^2+1}{M^4} + \left[\frac{(\gamma+1)^2}{4} + \frac{\gamma-1}{M^2} \right] \sin^2 \alpha \quad (160b)$$

$$d = -\frac{\cos^2 \alpha}{M^4} \quad (160c)$$

Equation (159) has three roots for $\sin^2 \sigma$, in general, of which the lowest corresponds to a decrease in entropy and thus to a physically impossible flow. The lower of the two remaining roots gives the desired weak-shock solution, while the uppermost root gives a strong-shock solution (e.g., a normal shock with $\sigma = \pi/2$ for $\alpha = 0$). Subroutine SHOCK first obtains the lowest of the three roots by solving equation (159) using Newton's method. With $u = \sin^2 \sigma$ and

$$F(u) = u^3 + bu^2 + cu + d \quad (161a)$$

$$F'(u) = 3u^2 + 2bu + c \quad (161b)$$

an improved estimate u' of the root is obtained, at each step of the iteration, using the formula

$$u' = u - \frac{F(u)}{F'(u)} \quad (162)$$

The convergence criterion is that $u' = u$ to within 1 part in 10^8 . If convergence is not achieved within 50 iterations, an error message is written and the RETURN is executed.

Use of an initial estimate $u = 0$ ensures convergence to the lowest root. Once this root u_1 has been obtained, the desired weak-shock solution is obtained by factoring the equation $F(u) = 0$:

$$(u-u_1) (u^2 + Au + B) = C \quad (163)$$

Equating the coefficients of corresponding powers of u in (161a) and (163) gives

$$A = u_1 + b \quad (164a)$$

$$B = -d/u_1 \quad (164b)$$

The two remaining roots of the cubic are then obtained by equating the quadratic factor in (163) to zero:

$$u = \frac{1}{2} [-A \pm \sqrt{A^2 - 4B}] \quad (165)$$

If the flow-deflection angle α is too large, the argument of the square root in (165) becomes negative and equation (159) has no physically valid solution. In this case, the error indicator IERR is set to 0 and the RETURN is executed. When $A^2 - 4B$ is positive, the lower of the two values (165), u_2 , is equal to $\sin^2 \sigma$ for the weak shock.

After the shock angle σ has thus been determined, subroutine SHOCK calculates the conditions behind the shock using perfect gas relations (ref. 15, p. 86). The only result required for use in NATA is the shock pressure ratio,

$$\frac{p_2}{p_1} = \frac{2\gamma M^2 \sin^2 \sigma - (\gamma - 1)}{\gamma + 1} \quad (166)$$

3.62 Subroutine SIMQ

SIMQ is a standard subroutine for solving systems of simultaneous linear equations. It was obtained from the IBM Scientific Subroutine Package (ref. /), and was modified slightly for use in NATA. The modifications consisted of converting it into a double precision routine and the addition of simple coding to allow SIMQ to handle the special case of one equation in one unknown. The only available documentation of SIMQ is that provided by comment cards in the source program.

3.63 Subroutine STUNTS

This routine performs two separate and unrelated operations. When called through the principal entry (STUNTS), it produces tables of species thermal properties as functions of temperature for the species in the specified gas model. When called through the entry STUNT2, it gives an edit of the transport property calculations, and optionally, a table and a punched deck containing averaged species cross section data.

3.63.1 Thermal Property Edit

The thermal property edit is produced by the DØ loop ending at statement 110. For each species J, the properties are computed at a preset series of temperatures TT(I) up to the input value of the reservoir temperature CTAP. These temperatures are as follows:

From 100°K to 1000°K at intervals of 100°K
 From 1000°K to 3000°K at intervals of 200°K
 From 3000°K to 10000°K at intervals of 500°K
 From 10000°K to 20000°K at intervals of 1000°K
 From 20000°K to 50000°K at intervals of 2000°K
 From 50000°K to 125000°K at intervals of 5000°K

1
The maximum number of temperatures is 74. (However, the thermo fits for the molecular species in the compiled-in gas models are useful only up to about 20000°K.)

The thermal properties are calculated by calling subroutine THERM for each temperature. THERM is programmed to calculate the properties using two different methods, the physical model and the thermo fit (Section 2.2 of Vol. I, ref. 1). The physical model is used for temperatures more than 500°K below the switchover temperature, CTMXX·CTAP. The thermo fit is used for temperatures more than 500°K above the switchover temperature. In a range 1000°K wide centered at the switchover temperature, a linear combination of the results from the two methods is used, to provide a continuous transition.

1
To obtain thermal properties based solely on the physical model, rather than on a mixture of the two methods, subroutine STUNTS first sets CTMXX to 1000, and calls THERM for each temperature. With this large value of CTMXX, all of the temperatures are more than 500°K below CTMXX times the reservoir temperature. The results of the thermal property calculations based on the physical model are stored in the array TPRØP(I,K,M) for M = 1. In the case of a nonstandard species for which physical model data have not been provided (ETAJ(J) = 0), the elements of TPRØP for M = 1 are all set to zero and the calculations based on the physical model are skipped.

For each species, if thermo fit data have been provided (IGJ(J) ≠ 0), CTMXX is set to -1000 and THERM is called again for each temperature up to the reservoir temperature, CTAP. With this negative value of CTMXX, all of the temperatures are more than 500°K above CTMXX·CTAP, so that the calculations in THERM are based solely on the thermo fit. The results are stored in TPRØP(I,K,M) for M = 2.

Finally, all of the results for the species are printed out. The properties computed and printed are

$$\text{TPR}\emptyset\text{P}(\text{I},1,\text{M}) = \frac{\mu_j^{\circ} - H_{j0}^{\circ}}{R_0 T}$$

(chemical potential in excess of the formation enthalpy, divided by $R_0 T$)

$$\text{TPR}\emptyset\text{P}(\text{I},2,\text{M}) = H_j - H_{j0}^{\circ}$$

(molar enthalpy in excess of the formation enthalpy, kcal/mole)

$$\text{TPR}\emptyset\text{P}(\text{I},3,\text{M}) = C_{pj}$$

(molar heat capacity, cal/mole $^{\circ}$ K)

$$\text{TPR}\emptyset\text{P}(\text{I},4,\text{M}) = S_j^{\circ}$$

(molar entropy, cal/mole $^{\circ}$ K)

Examples of the thermal property edit are shown in figures 19 to 46 of Volume II (ref. 2).

3.63.2 Transport Property Edit

The transport property edit consists of three parts, the first two of which are always obtained when STUNT2 is called. These parts are (1) an edit of the steps in the transport cross section calculation as specified in the input or in the precoded data; (2) an edit of the "edited" cross section calculations after unneeded steps have been deleted, and needed but unspecified steps have been added on the basis of default options; (3) an edit of the averaged pair cross sections as functions of temperature.

(1) Specified steps in the cross-section calculation.--

This first part of the transport property edit is implemented by the DØ loop ending at statement 150. The index L runs over the steps of the cross section calculation in the order in which they are to be performed. The sequencing array ISEQ(L) selects the steps from the input or precoded data; i.e., the step which is carried out first is the one indexed I = ISEQ(1), etc. NNQI is the number of species pairs to which the cross section calculated in the Lth step will be applied. The indices of the species belonging to such a species pair are IIQJ, JJQJ (referred to the

master list of species).^{*} For each step, STUNT2 prints out L, I, the index KKQ(I) specifying the type of cross section formula used in the step, the parameter values VV(K,I), and the species names for the first pair of species to which the results of the step are applied. The names of species for the remaining species pairs (if any) are then printed on successive lines.

(2) Steps in the edited cross section calculation.--
The DØ loop ending at statement 190 prints out a description of the steps in the "edited" cross section calculation which will actually be used in the transport property calculations for the current case. This "edited" calculation differs from the "input" calculation in two major ways: The steps applying only to species which are absent from the current gas model have been deleted, and steps which are required but which were not specified have been provided on the basis of default options** such as the cross section mixing rule.

The index L in the DØ loop ending at 190 runs over the steps in the edited calculation, in the order in which they are to be performed. KQ(L) is the index of the option (cross section formula) to be used in the Lth step. NV(KK) is the number of parameters used by this option. All of the parameters for all of the steps are stored, without gaps, in the singly dimensioned array V(I). The parameters for the Lth step occupy the locations from I = MV1 to MV2 in this array.

The indices of the species belonging to the species pairs to which the steps of the cross section calculation apply are stored, without gaps, in two arrays IQ(LQ), JQ(LQ). These indices IQ and JQ are referenced to the species list for the current gas model, rather than to the master list of species. The index LQ for the last species pair to which the results of the Lth step apply is NQ(L).

^{*}See Section 4.2 of Volume II (ref. 2).

^{**}See Section 4.6 of Volume II (ref. 2).

For each step L, STUNT2 first prints L, the option index KQ(L), and the parameter list V(I) for the option. Then it prints the names, HP(I), HP(J), for the first species pair using a carriage-control character + in column 1 to place these names on the same line. If there are any additional species pairs to which the step is applicable, their names are printed on successive lines.

(3) Averaged transport cross sections.--If ISWLB is negative, STUNT2 prints a table giving the transport cross sections for the various species pairs as functions of temperature. If ISWLB = -1, STUNT2 also punches these same data on cards for use by other computer programs. Some of the cross sections for charged species depend upon the electron partial pressure as well as the temperature. In STUNT2, the mole fractions SAVEC(I) used by the transport property routines are set to their values in the reservoir (for the current case), and are held fixed as the temperature is varied. The pressure is also held constant at its reservoir value. Thus, all of the cross sections printed are based upon the electron pressure in the reservoir.

The calculations and output are provided by the DØ loop ending at statement 230. For each species pair (I,J) with $I \leq J$, the temperature is varied from 1000°K to the reservoir temperature CTAP by intervals of 1000°K. At each temperature, subroutine TRANSP is called. With ISWLB < 0, TRANSP does not compute the viscosity, Prandtl number, etc., but does call lower-level transport routines to set up the cross section array Q(L,I,J), where I and J are the species indices and L runs from 1 to 3. These data are printed (and punched for ISWLB = -1), together with the temperature, species indices, species names, and a counter ICARD which serializes the punched cards. Their relation to the cross sections used in transport theory is (Section 3 of Vol. I, ref. 1):

$$\begin{aligned} Q(1,I,J) &= \bar{\Omega}_{ij}^{(1,1)} \\ Q(2,I,J) &= \bar{\Omega}_{ij}^{(2,2)} \\ Q(3,I,J) &= B_{ij} * \bar{\Omega}_{ij}^{(1,1)} \end{aligned}$$

They are expressed in units of 10^{-16} cm^2 .

3.64 Subroutine THERM

THERM is the routine which computes the species thermal properties: μ_j^0 (chemical potential at standard pressure), H_j (molar enthalpy), S_j^0 (molar entropy at standard pressure), and C_{pj} (molar heat capacity at constant pressure). In its internal calculations, NATA uses these properties in nondimensional form; the quantities actually computed in THERM are

$$\text{XMJAT}(J) = \frac{\mu_j^0}{R_0 T} \quad (167)$$

$$\text{SEJ}(J) = \frac{H_j}{R_0 T_0} \quad (168)$$

$$\text{SENT}(J) = \frac{S_j^0}{R_0} \quad (169)$$

$$\text{CCPJ}(J) = \frac{C_{pj}}{R_0} \quad (170)$$

where $J = j$ is the species index in the list of species for the current case. In these formulas, R_0 denotes the universal gas constant and T_0 the reservoir temperature.

When the subroutine is called through the entry THERM1, only the heat capacity is computed.

THERM provides two methods for calculating the species thermal properties: the "physical model" and the thermo fit. These techniques have been explained and discussed in Section 2.2 of Volume I (ref. 1). Data for the physical model are provided for all standard species. In the current version of NATA, thermo-fit data are provided only for the diatomic molecules and molecular ions in air and the planetary atmosphere models. The array IGJ(J) indicates whether thermo fit data are available for each species. If $\text{IGJ}(J) = 0$, there are no thermo fit data for the J^{th} species; in this case the physical model is used

for the species at all temperatures. If $ETAJ(J) = 0$, there are no physical-model data, and the thermo fit is used at all temperatures.

In the case of species for which both physical-model and thermo fit data have been provided, the physical model is used from low temperatures up to a nominal switchover temperature, $CTMXXI$, which is preset to $5000^{\circ}K$; and the thermo fit is used at higher temperatures. To prevent possible program failures due to discontinuities in computed species properties resulting from small mismatches between the physical model and thermo fit results, the switchover is spread over a temperature range $1000^{\circ}K$ wide, centered at $CTMXXI$. Equations I(58) are used to mix the results from the two techniques in this temperature range.

The temperature variable used by NATA in its internal computations is CT , a nondimensional temperature defined as T/T_0 . The corresponding nondimensional switchover temperature is $CTMXX = CTMXXI/T_0$. For species and temperatures for which the thermo fit is used, the species properties are computed from the formulas I(33), I(34), I(29), and I(31). Prior to the first call to *THERM*, the thermo fit coefficients $TFA(J) = a$, etc., have been nondimensionalized in subroutine *INIT* so as to allow direct use of the nondimensional temperature CT in the thermo fit formulas.

The physical model calculations are done in two steps. The first step computes the translational, rotational, and vibrational contributions to the species properties, and the second step the contributions due to electronic excitation. The first step is coded separately for monatomic and diatomic species, on the one hand, and for linear triatomic species, on the other. In the case of monatomic and diatomic species, the calculations are done using equations I(50), I(51), I(54), I(56), and I(29). For linear triatomic species, equations I(52), I(55), I(57), and I(29) are used.

If the control parameter *INEQV* is nonzero, the properties of molecular species are calculated on the assumption that the vibrational degrees of freedom are frozen at

the reservoir temperature. In this case, the formulas given in Section 2.4 of Volume I (ref. 1) are used to calculate the species properties.

3.65 Subroutine THRØAT

The function of THRØAT is to adjust the nozzle geometry in the downstream region so as to make the effective area ratio continuous between the upstream solution by the inverse method and the direct downstream solution. It performs this function by multiplying the rescaling factor RSA, which is used in GEØMAR, and which is initially 1.0, by the factor AG/S1. Here S1 is the geometric area ratio based on the specified nozzle geometry before the adjustment, and AG is the geometric area ratio calculated from the effective area ratio AFNX. If the boundary layer is not included in the solution; AG is equal to AFNX. If the boundary layer is included, AG is computed from AFNX by calling subroutine AGSØLN. Since boundary layer displacement effects upon the inviscid flow are neglected during the nonequilibrium solution by the inverse method, the rescaling also provides continuity of the effective area ratio when the coupling of the inviscid flow with the boundary layer is switched on following the call to THRØAT.

In addition to calculating RSA, THRØAT prints an informative message specifying the conditions at the current flow point, and resets the indicator IUPD from 1 to 0 to signify that the switch from the inverse method to the direct integration has occurred.

3.66 Subroutine TRANSP

Transport property calculations are performed in the NATA code under the general supervision of subroutine TRANSP. This subroutine contains two entries: TRANSX which is called to initialize the routines, and TRANSP(TTAB,P) which carries out the actual transport property calculations for any specified gas conditions.

The entry TRANSX is called prior to the start of actual transport property calculations, in order to inform the

routines of the species to be used in the calculations and to perform certain required initializations. The entry first sets the total number of species $N=ISS$ to be used in the transport calculations and then calls subroutine XSECT which selects cross section data for the species to be used in the calculations from the data available in the code and arranges them in a form convenient for the subsequent computations. Finally, TRANSX calculates a number of quantities depending on the species molecular weights which will be required in the subsequent transport computations, before returning control to the calling program. The quantities calculated here are

$$ZM1(I) = \frac{128 \times 10^{-24}}{9 \pi N_0 k^3} \quad W_i = 2.8567 W_i \frac{\text{cm}^2(^{\circ}\text{K})^3}{\text{milliwatts}^2(\text{\AA})^4} \quad (171a)$$

$$ZM2(I) = \frac{4}{15 N_0 k} \times 10^7 \quad W_i = 0.032064 W_i \text{ g } ^{\circ}\text{K/joule} \quad (171b)$$

$$C(I,J) = \frac{8 \times 10^{-12}}{3k} \sqrt{\frac{2 W_i W_j}{\pi N_0 k (W_i + W_j)}} \quad \text{for } I \leq J \quad (171c)$$

$$C(I,J) = \frac{W_i - W_j}{W_i + W_j} \quad \text{for } I > J \quad (171d)$$

Once the gas species have been defined by calling TRANSX, the transport properties for any given gas condition are computed by calling TRANSP(TTAB,P), where TTAB is the gas temperature and P is the gas pressure in atmospheres. The species mole fractions are obtained from the array SAVEC(J) in common block /TEMPRY/. The first step in the computation is the calculation of the molar specific heat for each species in the mixture. This calculation is carried out by calling the entry THERM1 of subroutine THERM. Input to the calculation is by means of the common variable CT which specifies the ratio between the temperature TTAB for which the specific heats are to be calculated and the reservoir temperature CTAP. The nondimensional molar heat

capacities at constant pressure $W_j c_{pj}/N_0 k$ for each species in the mixture are then returned to subroutine TRANSP in the common variable CCPJ(J). These values are saved by the routine in the array DH for later use in computing the internal thermal conductivity from equation I(95), and are also used to compute the nondimensional frozen specific heat $CPT\theta T = W c_{pf}/N_0 k$ for the mixture, which is required in the calculations of the Prandtl and Lewis numbers, equations I(83) and I(84). In calculating the specific heats, it is necessary to save the previous values of CT and CCPJ in a temporary location and then restore them for later use in other portions of the flow field calculations.

After completing the specific heat calculations, TRANSP next calls subroutine PUTQIN which computes the collision cross sections $\bar{\Omega}_{ij}^{(1,1)}$, $\bar{\Omega}_{ij}^{(2,2)}$, and $B_{ij}^* \bar{\Omega}_{ij}^{(1,1)}$ in units of 10^{-16} cm^2 for all pairs of species in the mixture at the specified temperature and pressure. These cross sections are returned to subroutine TRANSP in the upper half ($I \leq J$) of the common arrays $Q(1,I,J)$, $Q(2,I,J)$, and $Q(3,I,J)$ respectively. If an edit of pair cross sections has been requested ($ISWLB < 0$), a RETURN is now executed, and subroutine STUNTS then prints the cross sections contained in the Q array. However, in a normal NATA run ($ISWLB \geq 0$), subroutine TRANSP uses these values of the $\bar{\Omega}_{ij}^{(l,s)}$ to compute the matrix elements $A_{ij}^{(\alpha)}$, $a_{ij}^{(\alpha)}$ and $\Delta_{ij}^{(1)}$, equations I(88) to I(90), which are required in the transport property calculations, and stores them in the Q array, replacing the $\bar{\Omega}_{ij}^{(l,s)}$ values which were previously stored there. The statements down to 80 compute and store these quantities as follows:

$$Q(1,I,J) = \frac{1}{k} \sqrt{T} \Delta_{ij}^{(1)} \left(\frac{\text{cm } (^{\circ}\text{K})^{3/2}}{\text{milliwatt}} \right) \quad \text{for } I \leq J \quad (172a)$$

$$Q(1,I,J) = \frac{25}{4} \sqrt{T} a_{ij}^{(K)} \left(\frac{\text{cm } (^{\circ}\text{K})^{3/2}}{\text{milliwatt}} \right) \quad \text{for } I > J \quad (172b)$$

$$\begin{aligned} Q(2,I,J) &= \frac{25}{8} \sqrt{T} ZM2(I) A_{ij}^{(\mu)} \left(\frac{\text{cm } (^{\circ}\text{K})^{3/2}}{\text{milliwatt}} \right) \\ &= \frac{5}{6} \frac{\sqrt{T}}{k} \Delta_{ij}^{(2)} \quad \text{for } I \leq J \quad (172c) \end{aligned}$$

$$Q(2, I, J) = \frac{25}{8} \sqrt{T} a_{ij}^{(\mu)} \left(\frac{(^{\circ}K)^{1/2}}{\text{millipoise}} \right) \text{ for } I > J \quad (172d)$$

$$Q(3, I, J) = \frac{25}{8} \sqrt{T} A_{ij}^{(K)} \left(\frac{\text{cm } (^{\circ}K)^{3/2}}{\text{milliwatt}} \right) \quad (172e)$$

In connection with these computations, it may be noted that the diagonal matrix elements $a_{ii}^{(\alpha)}$ are not required in the transport calculations, as shown by equation I(86), so that it is unnecessary to compute them here. Since the quantity (172e) is not symmetric in i and j , it requires the whole $Q(3, I, J)$ matrix for storage, whereas the other matrix elements are all symmetric, and can be stored in half the array. The quantity $\rho c_p D_{ij} / \sqrt{T}$ in milliwatts/cm $(^{\circ}K)^{3/2}$ is also computed and stored by the code as FLEWIS, where D_{ij} represents the binary coefficient I(99) between the two species ID1 and ID2 which have been specified for the calculation of the Lewis number in the code input.

The calculation of the transport properties from the matrix elements (172) is begun in the DØ loop ending at statement 90, and is then carried forward in a series of three subroutines ELCØND, KINT, and KANDMU. The quantities returned to TRANSP by these subroutines are

$$\begin{aligned} \text{SIGMA} &= \sigma \text{ in mhos/cm} \\ \text{ZKINT} &= K_{\text{int}} / \sqrt{T} \quad (\text{milliwatts/cm-} (^{\circ}K)^{3/2}) \\ \text{ZK(1)} &= K_{\text{tr}} / \sqrt{T} \quad (\text{milliwatts/cm-} (^{\circ}K)^{3/2}) \\ \text{ZK(2)} &= \mu / \sqrt{T} \quad (\text{millipoise/} (^{\circ}K)^{1/2}) \end{aligned} \quad (173)$$

where σ , K_{int} , K_{tr} , and μ are respectively the electrical conductivity, the internal thermal conductivity, the translational thermal conductivity, and the viscosity of the mixture, as given by equations I(85) to I(87), I(95) and I(98). These quantities are then combined further by subroutine TRANSP according to equations I(83), I(84) and I(94) to obtain the viscosity VISC in poise, and the frozen

Prandtl number PRF and the atom-molecule Lewis number* FLEWIS. These quantities, along with the electrical conductivity SIGMA, are then returned to the calling program in the common block TRPROP. For ISW8B \neq 0, these data, as well as certain other quantities of interest, are also printed out by the final section of subroutine TRANSP before returning control to the calling program.

3.67 Subroutine WEDGE

Subroutine WEDGE calculates the pressure, heat flux, and boundary layer displacement thickness on wedge models. The calculations are done using the modifications I(501), I(503), I(505) of the results of the Cheng-Kemp theory. Optionally, the calculations are also done using the unmodified Cheng-Kemp theory if the input control variable ISW9B is set to a value in the range from -1 to -3 .

WEDGE obtains the free-stream flow properties required in the calculations from the array FVØUT in common block /ØUTPUT/. The data in this array refer to the current flow point, because the free stream conditions are always printed out immediately prior to a call to subroutine MØDEL which, in turn, calls WEDGE. These data are accessed by equivalencing the variable names used in WEDGE to the corresponding FVØUT array elements.

The subroutine logic is straightforward. All quantities which are independent of the angle of attack and the wedge leading-edge radius are computed first. The remaining calculations are performed and the results are printed out in a pair of nested DØ loops over the index IR for leading edge radii and the index IA for angles of attack.

*The Lewis number as given by I(84) is multiplied by an integer variable IAMBIP, which is equal to 1 if the two species for which the Lewis number is calculated are both neutral, and equal to 2 if one of these species is an ion. The factor of 2 takes account of the ambipolar nature of the diffusion of ions in a neutral plasma.

For each angle-radius combination, the subroutine performs calculations for all of the specified distances from the leading edge. These distances are specified in the input in two independent ways:

- (1) A uniform sequence of NWX points, with an initial value of WX1 and an increment DWX;
- (2) A list of arbitrary values in ascending order.

Subroutine WEDGE folds these two sequences together into a single sequence arranged in ascending order, using logic similar to that employed in subroutine NEXTMP.

WEDGE calls subroutine SHOCK to compute the static pressure ratio across the oblique shock, denoted by APR(IA). This quantity is used in calculating the angle of attack parameter Γ (denoted by CAPGAM) from equation I(501) in the modified Cheng-Kemp theory. After Γ and ζ (ZETA) have been computed, WEDGE calls subroutine WESOLN to obtain the approximate solution of the Cheng equation. The arguments of WESOLN are double precision.

The calculations based on the modified Cheng-Kemp theory are performed for ITH = 1. When these calculations have been completed at a given model point, if ISW9B is negative, ITH is set equal to 2 and all of the calculations are done over using the unmodified theory.

The input variable ISW9B is also used to control the output of some results of secondary interest. If IS9 = |ISW9B| is equal to 1, the shock ordinate YS is printed; if IS9 is equal to 2, the nondimensional coordinate ZETA is printed; and if IS9 is equal to 3, both YS and ZETA are printed.

3.68 Subroutine WESOLN

The function of subroutine WESOLN is to solve the Cheng equation I(480) for z , zz' , and $(zz')'$ when ζ and Γ are given. The solution is based on the approximation I(484) which relates the solution for general Γ to that for $\Gamma = 0$.

The latter is obtained from Cheng's analytical solution I(482), I(483) for $\zeta > 10^{-8}$. For small ζ , there is a great deal of cancellation of terms in I(482), so that many significant figures of accuracy are lost. This problem is kept under control by performing all of the arithmetic in WESOLN in double precision, and by using an analytical series solution for $\zeta \leq 10^{-8}$:

$$z = 1.65096 \zeta^{2/3} + 0.50869 \zeta^{5/6} - 0.0249 \zeta \quad (174a)$$

$$zz' = 1.81711 \zeta^{1/3} + 1.25975 \zeta^{1/2} + 0.14712 \zeta^{2/3} \quad (174b)$$

$$(\cdot zz')' = 0.6057 \zeta^{-2/3} + 0.62988 \zeta^{-1/2} + 0.09808 \zeta^{-1/3} \quad (174c)$$

Equation (174a) was derived by Boger and Aiello (ref. 16). Equations (174b) and (174c) can be obtained from (174a) by differentiation.

For $\zeta > 10^{-8}$, equation I(482b) is solved for λ using Newton's method. The parameter λ is denoted by XL. An initial trial value of λ is obtained from the rough analytical approximation

$$\lambda = zz' = (6 \zeta)^{1/3} + (3 \zeta)^{1/2} \quad (175)$$

which has a maximum error of about 12 percent. At each step in the Newton iteration, ζ is calculated from the current trial value of λ using equation I(482b), and an improved estimate λ' of λ is obtained from

$$\lambda' = \lambda - (\zeta - \zeta_i) \frac{d\lambda}{d\zeta} \quad (176)$$

in which ζ_i is the input value ZETA and, from I(480) with $\Gamma = 0$ and I(483),

$$\frac{d\lambda}{d\zeta} = \frac{d}{d\zeta} \left(z \frac{dz}{d\zeta} \right) = \frac{1 + \sqrt{zz'}}{z} = \frac{1 + \sqrt{\lambda}}{z} \quad (177)$$

where z is computed from I(482a). The convergence criterion of the iteration is that the ζ from I(482b) equal the input value to within 0.01 percent.

Once the correct value of λ for the solution with $\Gamma = 0$ has thus been found, the three outputs of the routine are computed as follows: zz' (denoted by ZZP) is obtained from equation I(491), $(zz')'$ (denoted by ZZPP) from I(434) and I(485), and z (denoted by Z) from I(492).

3.69 Subroutine XSECT

Subroutine XSECT provides NATA with the capability to perform transport property calculations for gas models which do not contain all of the species listed in the pre-coded cross section data, and in which the species may be listed in a different order. For this purpose the subroutine sets up a new set of data for the cross section computations in common block /TRANS7/ which gives the same values for the cross sections as the original input data in common block /TRANS2/, but which is in a more condensed format, and is referenced to the set of species included in the current gas model rather than to the larger set of species for which cross sections are defined in the pre-coded and input data. The edited data in common block /TRANS7/ are then used by the code in computing collision cross sections for the gas in the subsequent transport property calculations. The original cross section data are also retained unchanged in common block /TRANS2/, in order to allow the running of stacked cases with different sets of species.

Since several of the options which are used for computing cross sections in NATA (see Section 4.6 of Volume II, ref. 2) give values for the cross sections which depend on the values computed in previous steps of the computation, the edited cross section data in common block /TRANS7/ are set up to carry out the steps of the cross section computations in the same order in which they were specified in the

original data; however, the species references in the data are changed so that the computed cross sections will be indexed in the same way as the species used in the current gas model, rather than being indexed to the original precoded or input data. Further, cross sections for pairs of species not required in the transport calculations are omitted from the edited data, and any missing cross sections are supplied according to the default options described in Section 4.6 of Volume II. The procedures used by subroutine XSECT in preparing these edited data are described in the following paragraphs. It may be noted that several of the cross section options available in the code contain features which require special treatment in this process, so that the coding of the subroutine becomes rather lengthy, although the basic concept is simple. A schematic flowchart of the subroutine showing the major control sections is provided in figure 27.

3.69.1 Initializations

The first executable statement in subroutine XSECT sets the number of species N for which cross sections are to be computed equal to the number of species ISS in the current gas model. The next two statements set $IELEC = 1$ if electrons are to be considered in the transport calculations and $IELEC = 0$ otherwise. If electrons are included in the calculations, they must always be the first species.

The statements from " $JJ = 50$ " down through statement 30 establish the correspondence between the species in the current gas model and the master list of species,* which is used in indexing the cross section data supplied in common block /TRANS2/. These statements use the common variable $IS(L)$, which gives the index IS in the master species list of the L^{th} species in the current gas model (for $1 \leq L \leq ISS$), to prepare a new array $I(J)$ which gives the position I in the current model of the J^{th} species in the master species list, for each species included in the current gas model. Species in the master species list which are not included in the current model are filled into the $I(J)$ array in order following the species included in the current model, so that

*See Section 4.2 of Volume II (ref. 2).

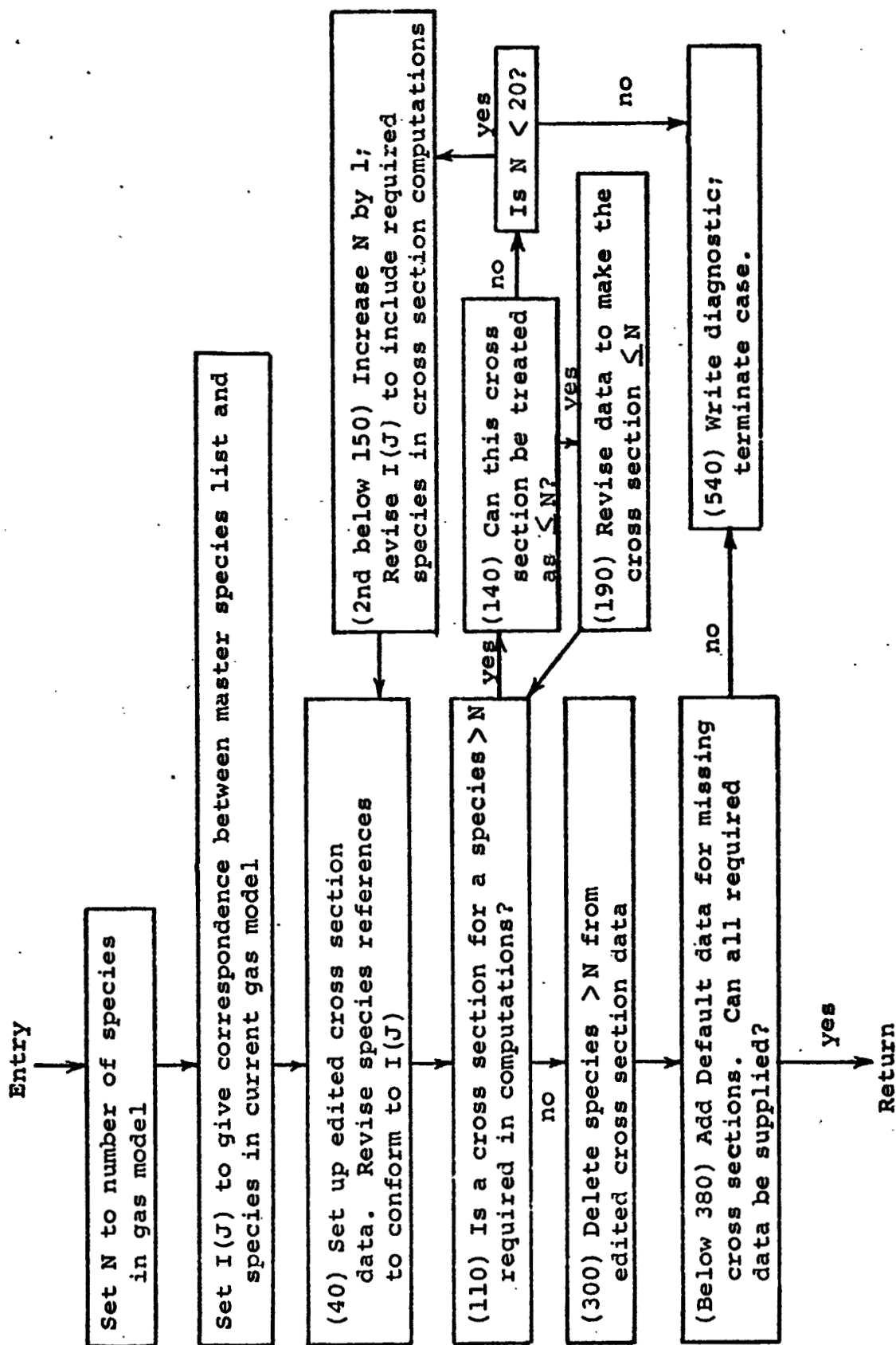


Figure 27. Schematic Flowchart of Subroutine XSCT

I(J) defines a rearrangement of the master species list such that all species J in the original master species list occur at some point I(J) in the rearranged list,* and the first ISS species in the rearranged list are just the species in the current model, in the correct order.

The three statements following statement 30 determine the indices ID1 and ID2 in the rearranged species list of the species to be used in the Lewis number calculations. They are set up so that $ID1 < ID2$ in all cases.

The common variable Q(1,J,L) is used as a flag in subroutine XSECT to indicate those pairs of species for which cross section data have been supplied. Statements 40 through 50 initialize this variable to 0 for all pairs of species included in the cross section calculations, and the value is then later set to 1 for each species pair for which data are provided. The value $Q(1,2,1) = 1$ is set initially to indicate that the pair (2,1) is used in the intermediate calculation of the cross sections in subroutine PUTQIN, and is thus not available for use in input. (Note that we are concerned here with ordered pairs (J,L). Only the pairs with $J \leq L$ are used in the actual transport property calculations,** while those with $J > L$ are available for intermediate calculations if desired.)

3.69.2 Resequencing of Steps and Revision of Species References

The coding following statement 50, down to statement 110, revises the species references in the input cross section data to conform to the rearranged species list I(J) described above and sets up a preliminary version of the edited cross section data in common block /TRANS7/ using these revised data. At this stage of the computation, the

*It is assumed that the master species list contains no more than 50 species; in the current version of NATA, array dimensions limit the number of species to 30.

**See the discussion of subroutine TRANSP in Section 3.66.

edited data contain the same information as the original cross section data in common block /TRANS2/, but the order of steps in the edited data is rearranged to conform to that specified by the sequencing array ISEQ, and the data on species pairs and cross section parameters for each step are close packed in the IQ, JQ, and V arrays. Further, $NNQ(L)$ in the edited data is taken to be the sum from 1 to L of the original $NNQ(L)$ values, and the indices of all species pairs in the edited data are changed to refer to the positions of the species in the rearranged species list, $I(J)$. Since only cross sections $Q(J,K)$ with $J \leq K$ are used in the transport property calculations, the correspondence between species pairs used in setting up the edited cross section data must preserve this property. Accordingly, the pair (J,K) in the original data is replaced by (I(J), I(K)) if J-K and I(J)-I(K) have the same sign and by the pair (I(K), I(J)) if they have different signs. Since this correspondence is 1-1 between ordered pairs of species, it is clear that when every species pair in the original cross section data is replaced by its corresponding pair in the edited data, the calculated cross section values will all be the same as before, but will be rearranged in the order required for the transport property calculations.

The above correspondence between species pairs is carried out in the NATA code by subroutine CXSECT, which returns the revised indices of a species pair in its argument list when it is called with the original indices. In the coding down through statement 80, this correspondence is used to set up the IQ, JQ list of species pairs for the revised cross section computations. This section of coding also sets $Q(1,L,J) = 1.$ for those species pairs which are included in the IQ, JQ list.

For the cross section options KQ = 9 and 12, there are also references to species pairs in the V parameters of the option which must be revised. This revision is carried out by the statements following statement 80 down through the statement before 110. Subroutine BXSECT searches through the steps L of the cross section computation in common block /TRANS7/, starting with the first step after the value of L specified in its argument, and looking for

a step with $KQ(L) = 9$ or 12 . When such a step is found, the subroutine returns the index of the step L , the index $MMV + 1$ of the first V parameter for the step, and the indices LL and J of the species pair which is referenced in the V parameters of the step. Subroutine $XSECT$ then calls $CXSECT$ to revise the indices for the pair and goes on with the search by calling $BXSECT$ again. This process is continued until the final step $L = NKQ$ of the computations is reached; at this point $BXSECT$ sets L to 0 as a signal that the search has been completed and control passes to the next section of code (statement 110). In addition to revising the indexing of the species pairs, this section of code also sets MMV , for later use, equal to the total number of V parameters used in the cross section computations.

3.69.3 Deletion of Unused Steps

When control reaches statement 110 of subroutine $XSECT$, the rearrangement of the species in the cross section data to conform to the order in the current gas model has been completed. The following section of coding, from statement 110 through 380, is now concerned with increasing the efficiency of the cross section computations by omitting unnecessary steps in the computation. In this procedure, the statements from 110 to 300 determine which cross sections are to be omitted from the computations, while those from 300 through 380 carry out the actual editing of the cross section data to delete these values.

As noted previously, the number of species N to be included in the cross section computations is set equal to the number of species in the current gas model at the beginning of subroutine $XSECT$. Thus only cross sections between pairs of species with indices $I \leq N$ in the revised cross section data are required in the transport property calculations. Ordinarily cross sections involving a species with index $> N$ can be omitted from the cross section computations without changing the values of the computed cross sections for the remaining species with indices $\leq N$; however, for the options $KQ = 9$ and 12 used in the cross section computations, it may happen that the cross section values

for some pair of species with indices $\leq N$ are to be computed from values obtained previously for some other species pair involving a species with index $> N$. Statements 110 to 140 check for this possibility, again using subroutine BXSECT to search for those steps in the computation with $KQ = 9$ or 12 . (Note that the coding here assumes that $L = 0$ when statement 110 is reached. This is always true, since statement 110 can only be reached from the IF statements following statements 100 and 290.) If this search does not reveal any pair of species with an index $> N$ which affects the cross section computations for species with indices $\leq N$, then it is possible to omit all species with indices $> N$ from the cross section computations without affecting the values of the calculated transport properties, and control passes at once to statement 300. If, on the other hand, some species pair (LL, J) with an index greater than N is found which affects the computations for indices $\leq N$, then the search for further such pairs is temporarily discontinued and the code immediately begins looking for a way to include the pair (LL, J) in the computations. The first approach considered is to treat the cross sections for the pair (LL, J) as an intermediate step in the computation of the cross sections for the species with indices $\leq N$, and to fill the computed cross sections for this pair into some unused location (II, JJ) of the cross section array for species $\leq N$, where $JJ < II \leq N$.^{*} Statements 140 to 150 look for such a location, using the previously set array $Q(1, II, JJ)$ as a flag to indicate which locations are available.

If a suitable location for intermediate storage of the pair (LL, J) is not found in statements 140 through 150, then the following section of code down to statement 190 increases the number N of species to be included in the cross section computations by one, rearranges the list of species $I(J)$ for the edited cross section data so that the species corresponding to the maximum index of the pair (LL, J) will be the new species included in the cross section computations (note that by definition this species was not included

^{*}Remember that only species with $II \leq JJ$ are used in the transport calculations.

in the computations previously), and then returns control to statement 40 to begin the preparation of the edited cross section data again from the beginning, using the new arrangement of species I(J). If this procedure would lead to a value of $N > 20$, however, so that the dimensions allowed for the cross section computations in the code would be exceeded, it is not carried out and control passes instead to Statement 540 where a diagnostic message is printed and the error indicator ERR is set to terminate the case. If more than 50 species were specified in the master species list, the case may also be terminated by reaching the GØ TØ 540 statement after statement 180.

When the statements from 140 through 150 are successful in finding an unused location (II,JJ) which can be used for storing the cross sections of the species pair (LL,J), control passes to statement 190, where the value of Q(1,II,JJ) is immediately set to 1.0 to indicate that the pair (II,JJ) is now being used in the cross section computations. The following statements down to statement 300 then revise the edited cross section data in common block /TRANS7/ to change all references to the pair (LL,J) to (II,JJ). To carry out this revision without changing the values of the computed cross sections, any steps in the computation which use the pair (LL,J) in the empirical mixing rule $KQ = 10$ must also be converted in the revision process to use the generalized mixing rule $KQ = 12$. This conversion is carried out by the coding down through the statement $NKQ=NSV$ after statement 270. The code first searches through all steps L of the cross section computations to find any steps using the option $KQ = 10$ which refer to the pair (LL,J). This may occur in two ways: either the species pair (LL,J) may be a non-diagonal pair (i.e., $LL \neq J$) which is computed with the option $KQ = 10$ at some step of the computations, or it may be a diagonal pair (i.e., $LL = J$) which is used in the option $KQ = 10$ to compute the cross sections for some non-diagonal pair involving the species LL. Both of these possibilities are checked for by the IF statement below statement 200, and if either is found the following statements then revise the edited cross section data in common block /TRANS7/ to convert the computation for the given species pair to the option $KQ = 12$. This is accomplished

by inserting the required data for the $KQ = 12$ option into the KQ , NQ , IQ , JQ and V arrays just ahead of the data for the step L which originally computed the cross section, and moving the remaining data in these arrays down to make room for the new data. The original option $KQ = 10$ is retained in the data immediately following the new $KQ = 12$ option, but is no longer applied to that species pair for which the $KQ=12$ option is being used. The values of MMV , NKQ and L are also adjusted to agree with the revised data. In these computations, subroutine $BXSECT$ is used to locate the position MM in the V array at which the new data are to be inserted. This is done by replacing NKQ by L in common block $/TRANS7/$, so as to cause $BXSECT$ to end its search of the KQ array, when step L is reached. The true value of NKQ is saved in the variable NSV throughout this section of the code and is restored to its proper location at the end of the section.

After the required conversions from $KQ = 10$ to $KQ = 12$ have been made, control passes to the section of code between card $XSE\ 213$ and statement 300 , which changes all references to pair (LL,J) in the IQ , JQ , and V arrays to refer to the pair (II,JJ) instead, using the same algorithms as were used previously in rearranging the species in statements 40 through 110 . When these changes have been completed, control returns again to statement 110 and the code again searches the edited cross section data for a pair of species with an index $> N$ which affects the cross section computations for species with indices $\leq N$. If such a pair is found, it is eliminated from the edited cross section data as described above, and the whole process is repeated until either all species pairs with indices $> N$ which affect the cross sections for species $\leq N$ have been eliminated from the data and control passes to statement 300 , or else the procedures used in the code are unable to eliminate a species pair and cause the case to be terminated as described above.

By the time control reaches statement 300 in subroutine $XSECT$, the edited cross section data have been put into such a form that all cross section computations involving species with indices greater than N can be omitted

without changing the values of the computed cross sections for the species with indices $\leq N$. The revision of the data to omit these computations is carried out by statements 300 to 380. The pairs of species for which cross sections are computed at each step of the computations are given in the IQ and JQ arrays, so that the steps to be deleted from the computations can be determined simply by searching the IQ, JQ list for pairs in which at least one of the indices IQ or JQ is $> N$. This search is performed by the preliminary loop ending at statement 310, and the value of IQ for each pair which is to be deleted is set to 0 as a signal for the subsequent calculations.

To determine the required revision of the KQ, NQ, and V arrays in the edited data, it is necessary to establish the correspondence between these arrays and the IQ, JQ arrays. This is done and the arrays are revised in the loop over the steps L of the cross section computation which ends at statement 370. For each step L of the computations, the small loop ending at statement 320 determines the correspondence to the IQ, JQ list and calculates the total number of species pairs NEWLQ which are not deleted up through step L. This number is the new value of NQ for the step.

The second statement below 330 is the test to determine whether the current step L of the computations will be omitted from the revised data. The test compares the current value of NEWLQ with the value for the previous step; if these two values are equal, it indicates that the current step of the computations will not apply to any species pairs in the IQ, JQ list after the deletions are made, so that the step can ordinarily be omitted from the revised data without affecting the computed cross sections. However, for the option, KQ = 10 the step is not omitted, even though it does not apply to any species, since it may be required subsequently in supplying the default data for the cross sections, as described below.

The edited cross section data for those steps which are not to be omitted from the computations are placed in their proper locations in the revised KQ, NQ, and V arrays by the coding from the third statement below 330 down to the

statement before 360. Since the revised arrays in this stage of the calculations are always shorter than, or at most equal to, their initial length, this revision can be carried out entirely within the arrays themselves, without disturbing subsequent steps in the revision. The indices L and MV here indicate the positions of the data in the initial arrays, while NEWL and NEWMV are their positions after the desired omissions have been made. It may be noted further in connection with this calculation that no data need be added to the revised V array for $KQ = 10$, since this option requires no values for V in its operation. Also the number of V parameters $NV(K)$ required by each option is obtained from preset data in the code. For steps which are omitted from the computations, of course, no data need be added to the KQ, NQ, and V arrays; however, the total number of steps NKQ in the computations is reduced by 1 by statement 360.

Once the revision of the KQ, NQ, and V arrays has been completed, the revision of the IQ and JQ arrays is completed in the loop ending at statement 380 by deleting the unwanted species pairs and moving the remaining pairs up to fill in any resulting gaps. This process completes the editing of the cross section data to remove unnecessary steps in the computations which was begun at statement 110 of the subroutine, and provides a set of edited cross section data which will give the same computed cross sections as the original input data for all species included in the current gas model, but which includes only data for the N species for which cross sections are to be calculated, and omits all data for other species which may have been included in the input or the precoded data.

3.69.4 Default Options

The final section of subroutine XSECT, following statement 380, is for the purpose of providing cross section data according to the default options discussed in Section 4.6 of Volume II (ref. 2) for any pairs of species included in the transport calculations for which data were not supplied in the input. These data are added to the edited cross section data in common block /TRANS7/ to provide a complete set of data for the later cross section computations in subroutine PUTQIN.

The first section of coding below statement 380 supplies data for computing the unspecified neutral-neutral cross sections in the gas mixture from the empirical mixing rule, equation I(102). These data are added to the edited cross section data at the last step in which the option KQ = 10 is used, or, if this option is not used at all in the data, after the last step of the specified data. This procedure has been adopted in the code to permit data on the resonant contributions to the neutral-neutral cross sections to be specified after the last step in the computations using KQ = 10, without affecting the values of the default cross section computed from the mixing rule I(102).

The coding from statement 380 down to statement 400 determines the step LL in the cross section computations at which the unspecified neutral-neutral cross sections are to be added to the data. If no step with KQ(LL) = 10 is found in the data, an additional step is added for this purpose at the end, and the value of NQ(LL) for this step is set to NQ(LL-1), since as yet, the step does not apply to any pairs of species in the IQ, JQ list. The statements from 400 through 420 then determine the charge IZ, in units of the electronic charge, for each of the species in the transport calculations, using data from the common array LPIJ containing the α_{ij} matrix, and test the value of IZ to determine which are the neutral species. When no electrons are present in the gas, (i.e., IELEC = 0), all species are assumed to be neutral and IZ is not set. Since the correspondence I(J) which was previously established between the master species list for the code and the species in the transport property calculations is now no longer needed, the array I is now used to store the indices of the neutral species for later reference. Since no default option has been established in the code for the computation of cross sections for the collisions of a neutral species with itself, the code tests each neutral species to determine whether these data have been specified in the input and if they have not, terminates the case by transferring control to statement 540.

Once the neutral species have been determined, the code calls subroutine AXSECT to add the required data for these species to the edited cross section data in common block /TRANS7/. Subroutine AXSECT then takes the list of neutral species stored in the I array and tests all pairs (J,K) with $J \neq K$, using the previously set values of the common variable $Q(1,J,K)$, to determine those pairs for which cross section data have not previously been supplied. All such pairs found in this search are then added to the IQ, JQ list at the LLth step of the cross section computations, the values of the $Q(1,J,K)$ for the added pairs are set to 1.0, and the IQ, JQ, and NQ lists are appropriately revised to take account of these additions. In these calculations subroutine AXSECT assumes that the species to be tested are stored in numerical order in the I array, and that NQ(LL) has been properly set prior to entrance into the subroutine.

After completing the specification of the neutral-neutral cross sections, the code next goes on to add data specifying the unspecified ion-ion cross sections to the edited cross section data. The three statements immediately preceding statement 430 establish a location in the edited data for the insertion of these cross sections. If the last step NKQ in the current version of the data is not presently being used in the cross section computations, it is used for the new data, while if it is being used, an additional step is added to the computations at the end of the current data for the insertion of the new data.

The loop from statements 430 to 480 adds data for computing effective Coulomb cross sections $I(100)$ for the unspecified ion-ion collisions to the edited cross section data. The coding of this loop is set up to cause control to pass through the block of statements from 440 to 460 five times. For the case in which electrons are present (NELEC = 1), the values of the indices at each passage

through this block are shown in Table III.* For each passage through the block, the loop ending at statement 450 stores some of the species included in the transport property calculations in the array I, the exact species which are chosen depending on the values of the indices for that passage. Thus, referring to Table III, the array I consists of all singly charged ions in the transport calculations for the first passage through the loop, all singly charged ions plus electrons for the second passage, and so forth. The call to XSECT following statement 450 then finds all pairs of the species in the I array for which cross sections have not been previously specified, and revises the edited cross section data to include these pairs in the NKQ^{th} step of the computations. After returning to subroutine XSECT, the code checks to see whether any unspecified pairs were actually found by subroutine AXSECT in this search, and, if they were, supplies values of the parameters KQ and V for the step which are appropriate for the effective Coulomb potential $I(100)$ from the preset array $VC\emptyset UL$, and then adds another new step at the end of the computations to prepare for storage of the next set of cross section data. The number of entries $NEWMV$ in the V array, which was set originally in the loop ending at statement 340, is also revised to conform to the new data. Thus, the first passage through statements 440 to 460 sets the cross sections for all unspecified pairs of singly-charged ions to the effective Coulomb cross sections given for singly-charged ions in equations $I(100a)$. The

*Note that for $IELEC = 0$, all species are assumed to be neutral so that all the unspecified cross sections were previously set in treating the neutral-neutral interactions. Thus, no cross sections remain to be set in the present loop and the edited cross section data are accordingly unaffected by the loop, independent of the values of the indices.

TABLE III

INDICES USED FOR CALCULATION OF EFFECTIVE COULOMB
CROSS SECTIONS IN SUBROUTINE XSECT

Passage number through loop	J	KK	JJ	IE	Species in I array*	Species pairs for which cross section data are supplied*
1	1	1	1	1	1	(1,1)
2	1	1	2	0	e,1	(e,e) (e,1)
3	2	2	1	1	2	(2,2)
4	2	1	1	1	1,2	(1,2)
5	2	1	2	0	e,1,2	(e,2)

*The notation e indicates electrons, 1 indicates singly charged ions, and 2 indicates doubly charged ions.

second passage then sets the cross sections for all unspecified pairs involving either singly-charged ions or electrons; but, since the ion-ion cross sections were already set in the previous passage, only electron-electron and electron-ion cross sections are actually set in this passage. From equations I(100a) one sees that the effective Coulomb cross sections for electron-electron and electron-ion collisions are the same (for singly charged ions), so that both types of cross sections can be set in a single step of the computations. Similarly, the last three passages through the loop set the unspecified cross sections involving doubly-charged ions to the values indicated in equations I(100a), as shown in Table III. Thus, when all five passages through the loop have been completed, cross sections will have been specified in the edited cross section data for all pairs of charged species to be included in the transport calculations, up through doubly ionized species. Further, it should be noted that the last step NKQ in the edited data at this stage of the calculations will be an empty position which is not used in specifying the cross sections but is available for the addition of further data.

The preset coding of subroutine XSECT does not provide any default option for specifying the cross sections of ions which are triply charged or higher. Thus the deletion of code following statement 480 down through the IF statement below statement 490 checks for unspecified ion-ion cross sections for species with higher than double ionization, and, if any are found, terminates the case. Otherwise, the next three statements below the IF statement add all so far unspecified pairs of species in the transport property calculations to the final step NKQ of the cross section computations. Since all neutral-neutral and ion-ion cross sections have previously been specified, the cross sections added at this step are necessarily all neutral-ion cross sections. If any such cross sections are added, parameters are then added to the KQ and V arrays to compute them from an inverse fifth power potential ($KQ = 6$), using the data on the magnitude of the cross sections stored in the array locations OMEGA1(996), ASTAR(996), and BSTAR(996) in common block/TRANS4/. For the preset data used in the code, this results in the cross sections for the added pairs being set equal to the corresponding $N-O^+$ cross

sections. On the other hand, if no unspecified neutral cross sections are found, these data are not added and the value of NKQ is reduced by one to eliminate the unused step at the end of the cross section computations.

The above calculations complete the preparation of the edited cross section data in common block /TRANS7/. The number of steps NKQ in the computations is now checked, and if it does not exceed the maximum allowed by the program dimensions (NKQ = 100), control is returned to subroutine TRANSP. Otherwise the case is terminated by a transfer to statement 540.

In their final version, the edited cross section data prepared by subroutine XSECT specify the procedures to be used in computing all of the cross sections required for the transport property calculations in a given case, and are in a form suitable for later use by subroutine PUTQIN and its associated subroutines in the actual computation of these cross sections. These data are in their final form when control returns from subroutine XSECT, and are not altered by the code in the subsequent transport property calculations until XSECT is called again to prepare a new set of edited cross section data for another case.

4. GLOSSARY OF FORTRAN SYMBOLS

This section presents a complete glossary of Fortran symbols for the NATA code. Symbols appearing in common are listed first, followed by the remaining symbols in individual routines. The order is as follows: unlabelled common, labelled common in alphanumeric order of the block names, main program, subroutines and functions in alphanumeric order. The variables in each common block are listed in the order in which they appear in the block. The non-common variables are listed in alphanumeric order for each routine. The mathematical notation is as defined in Volume I (ref. 1). The list for unlabelled common includes some variables which are equivalenced to variables in unlabelled common.

4.1 Unlabelled Common

AA(I,J)	Matrix of coefficients and constants for a system of simultaneous linear equations
AAA(I,J)	Equivalent to AA(I,J)
CDIJ(I,J)	Matrix of coefficients $\bar{v}_{i-c,j}$ for expressing the dependent species in terms of the c independent species
CAPX(J)	Mole fraction X_J for the J^{th} species
GJ(J)	Species concentrations in moles per gram of mixture; equivalenced to CAPX(J)
A	α , constant in the density-area relation I(383)
AFNTS	Effective area ratio calculated for equilibrium flow in the nonequilibrium solution by the perturbation method
AFNX	Effective area ratio, A_e
AMACH	Mach number

AR	Indicator (initially 0) which is set to 1 when $d \ln \tilde{A}_e / dx$ becomes positive downstream of the throat in the nonequilibrium solution by the inverse method.
ARBA	Maximum allowable number of tries at switching from upstream to downstream region in nonequilibrium solution
ARBB	Counter for number of times upstream-downstream switching point is moved downstream
BZERØ	b_0/w^0 , where b_0 is the covolume of the molecules in the high-density modification of the equation of state and w^0 is the molecular weight of the undissociated gas.
C	Constant C in density-area curvefit relation I(383)
CARB	Effective area ratio at the beginning of the current integration step
CH	Nondimensional specific enthalpy, hw_0/R_0T_0 , where h = specific enthalpy, w_0 = reservoir molecular weight, R_0 = universal gas constant, T_0 = reservoir temperature
CHA	Nondimensional specific enthalpy in the reservoir
CLNT	$\ln T$, where T = temperature in $^{\circ}K$
CM	Molecular weight of gas mixture, W (g/mole)
CMA	Molecular weight in reservoir, w_0 (g/mole)
CRA	R_0 , universal gas constant (1.9872 cal/mole $^{\circ}K$)
CRP	R_0T_0 , where T_0 = reservoir temperature ($^{\circ}K$)
CRRB	Intermediate variable in calculation of entropy

CRS	Dummy variable for entropy
CSTA	$0.5 \ln (W_0/R_0 T_0)$, where W_0 is the reservoir molecular weight (g/mole), R_0 the universal gas constant, and T_0 the reservoir temperature ($^{\circ}\text{K}$)
CT	Ratio of the temperature to the reservoir temperature, T/T_0
CTAP	Reservoir temperature, T_0 ($^{\circ}\text{K}$)
CTB	Dummy variable for CT
CTC	Dummy variable for CT
CTMAX	Ratio of throat temperature to reservoir temperature
CTMXX	Temperature for switching from thermo-fit to physical model for species properties, divided by reservoir temperature
CTP	Temperature in $^{\circ}\text{K}$
CTPL	$\ln T_0$
CTT	CT at the last printed step of the non-equilibrium solution
CX	Streamwise coordinate $x(\text{cm})$, zero at the throat and positive downstream
CXB	CX at the beginning of the current integration step
CXMAX	Maximum allowable value of $x(\text{cm})$
DATEST	Value of effective area ratio at which nonequilibrium solution is switched from upstream to downstream region
DBTEST	(Not used)

DELT1	Nondimensional temperature decrement used in frozen and equilibrium solutions
DELT2	(Not used)
DELTAX	Increment in x (cm)
DLØGA	$d \ln A_e / dx$
DLØGR	$d \ln (\rho / \rho_0) / dx$, logarithmic derivative of nondimensional density
DT	$d(T/T_0) / dx$, gradient of nondimensional temperature
ENT	Intermediate variable used in frozen flow solution
FLUX	Nondimensional mass flux, $(\rho / \rho_0) \cdot u / \sqrt{R_0 T_0 / W_0}$
HDELX	DELTAX/2
PCT	$\delta(T/T_0)$, perturbation in nondimensional temperature
PCTEST	Tolerance on C_x when testing size of $\delta \chi_i$
PRES	Nondimensional pressure, p/p_0
PRESA	Reservoir pressure p_0 (atm)
PRESB	Nondimensional pressure at the preceding flow point; used in Mach number calculation
PRESTH	Nondimensional pressure at throat
PRHØ	Perturbation in nondimensional density, $\delta \rho / \rho_0$
RHAP	Density in reservoir, ρ_0 (g/cm ³)

RHØ	Nondimensional density, ρ/ρ_0
RHØB	RHØ at the preceding flow point; used in Mach number calculation
RHØBAR	Effective density in high-density equation of state divided by effective reservoir density
RHØC	(Not used)
RHØP	Density ρ in gm/cm ³
RHPL	$\ln \rho$
RHTH	Nondimensional density at throat, ρ_*/ρ_0
RØBARA	$\tilde{\rho}_0/\rho_0$
RØBARP	Effective density in reservoir, $\tilde{\rho}_0$ (gm/cm ³)
SCPG	$\sum_j \gamma_j c_{pj}/R_0$
SDT	Change in nondimensional temperature in integration interval
SEN	Entropy in cal/g °K
SHPG	$\sum_j \gamma_j H_j/R_0 T_0$
SC	Factor by which integration interval is increased
SL	Characteristic length, ℓ , now set at 1 cm
SL64	$1/\sqrt{F}$, where F is the conversion factor from calories to ergs
SM	Nondimensional mass flux at the throat, $RHTH \cdot (SU)_*$
SU	Nondimensional velocity, $u \sqrt{W_0/R_0 T_0}$

SU2	$(SU)^2$
SUMG	$\sum \gamma_j$
TEST	Convergence criterion value for Newton-Raphson iterations in calculations of thermochemical equilibrium
TESTB	Tolerance allowed in locating temperature at throat for frozen and equilibrium flow
TPRINT	Nondimensional temperature interval at which results of nonequilibrium calculation are to be printed
TSTØP	Minimum value of nondimensional temperature desired in solution
UP	(Not used)
ZP	Natural log of nondimensional pressure
ZPA	Natural log of reservoir pressure in atm
BE(I)	$\beta_i = \sum_j \beta_{ij}$
BET(I)	$-1 + \nu^*_{i-c}$
BLBK(I)	Temporary storage area for boundary layer properties at the switch point from the upstream to downstream solutions
CAI(I)	A_i , constant factor in reaction rate formula
CAPQ(I)	Number of gram atoms of the I^{th} chemical element per mole of the cold gas; equivalenced to BLBK(I)
CAPXTH(J)	X_{j*} , mole fraction of J^{th} species at throat
CCI(J)	Temporary storage used in calculation of species molecular weights; equivalenced to CAPXTH(J)

CCPJ(J)	Molar heat capacity of the J th species divided by R ₀
CEACT(I)	Activation energy for i th reaction (cal/mole °K)
CGI(I)	Species molecular weights (g/mole)
CGMU(I)	$\ln X_i$, logarithms of mole fractions
CHI(I)	χ_i , variable measuring the departure of the i th reaction from equilibrium, eq. I(289)
CHII(I)	Intermediate variables in computation of equilibrium constant based on mole fractions
CLN1MC(I)	$\ln (1 - \chi_i)$
CLNPI(I)	$\ln P_i$, variables in nonequilibrium solution
CMW(I)	Atomic weights of elements (g/mole)
DGJ(J)	$d \gamma_j / dx$; equivalenced to GJA(J)
ETAI(I)	η_i , temperature exponent in rate constant for i th reaction
ETAJ(J)	n_j , number of atoms in a molecule of the J th species; 0 value indicates no physical model data for thermal properties
GJA(J)	γ_{j0} , concentration of j th species in reservoir
GJB(J)	GJ(J) at the beginning of the current integration step
PERTGJ(J)	$\delta \gamma_j$, perturbations in species concentrations
PGJ(J)	Intermediate variables in Newton-Raphson calculation of equilibrium composition.

PI(I)	Variables P_i in nonequilibrium solution, eq. I(288)
PICHI(I)	$P_i \chi_i$
QM(I)	Number of moles q_j of the j^{th} independent species in one mole of mixture
QQ(I)	1 if a third-body list is provided for the I^{th} reaction, 0 if not
SAJ(J)	$b_j + (1.5 + n_j) \ln T_0$, constant in physical-model expression for μ_j^0
SBJ(J)	b_j , eqs. (51), Vol. I; equivalenced to SAJ(J)
SDCHI(I)	$\delta \chi_i$, perturbation in χ_i
SDGJ(J)	Change in γ_j over an integration step
SENT(J)	Nondimensional species entropy, S_j^0/R_0
SHJ(J)	Nondimensional species enthalpy, $H_j/R_0 T_0$
SHJA(J)	Nondimensional formation enthalpy of j^{th} species, $H_{j0}^0/R_0 T_0$
SHJAP(J)	Formation enthalpy of j^{th} species, cal/mole
SKIL(I)	Intermediate variable in calculation of dependent species concentrations from concentrations of independent species
SS(J)	$\ln \gamma_{j0}$, logarithms of reservoir species concentrations
TB(J)	Temporary storage for quantities necessary to restart a step in the nonequilibrium calculation
TFA(J)	a_j , coefficient in thermo-fit expressions for H_j and μ_j^0

TFB(J)	b_j , coefficient in thermo-fit expressions for H_j and μ_j^0
TFC(J)	c_j , coefficient in thermo-fit expressions for H_j and μ_j^0
TFD(J)	d_j , coefficient in thermo-fit expressions for H_j and μ_j^0
TFE(J)	e_j , coefficient in thermo-fit expressions for H_j and μ_j^0/T
TFK(J)	k_j , coefficient in thermo-fit expressions for H_j and μ_j^0
THEV(J)	Nondimensional characteristic vibrational temperature, θ_{vj}/T_0
THEVP(J)	Characteristic vibrational temperature for j^{th} species, θ_{vj} ($^{\circ}\text{K}$)
XMJAT(J)	μ_j^0/R_0T
XNUI(I)	$\nu_i = \sum_j \nu_{ij}$, sum of stoichiometric coefficients on the reactant side of the i^{th} reaction
BETA(I,J)	$\beta_{ij} = \nu'_{ij} - \nu_{ij}$
BTA(I,J)	Equivalenced to BETA(I,J); not used
ELJ(L,J)	Nondimensional energy of L^{th} electronic level of j^{th} species, E_{lj}/R_0T_0
GELJ(L,J)	Degeneracy of L^{th} electronic level of j^{th} species, g_{lj}
XNUIJ(I,J)	ν_{ij} , stoichiometric coefficient of j^{th} species on reactant side of i^{th} reaction
XNUIJP(I,J)	ν'_{ij} , stoichiometric coefficient of j^{th} species on product side of i^{th} reaction
IC	Number of ions included in chemical model

IM	Index of the first species in the equilibrium flow calculation, incremented when electrons are dropped from calculation
INEQ	Indicator (initially 0), reset to 1 when numerical integration of nonequilibrium solution is begun
INEQV	Indicator for selecting whether equilibrium (0) or frozen (1) vibrational model is to be used
IP	(Not used)
IRUN	Run number for identification
ISC	Number of elements in mixture (including the electron, if ionized species are included)
ISCP1	ISC+1
ISMC	Number of species minus number of elements (n-c)
ISMCNR	Current value of (n-c) in equilibrium flow calculation (decreased by IC when electrons are dropped from the calculation)
ISR	r, number of reactions
ISS	n, number of species
ISSNR	Current value of n in equilibrium flow calculation (decreased by IC when electrons are dropped from calculation)
ISSP1	n+1
ISSP2	n+2
ISSP3	n+3
ISSP4	n+4

ISW1A	Input control variable; 0 suppresses frozen solution
ISW1B	Input control variable; nonzero value gives edits of transport cross section calculations
ISW2A	Input control variable; 0 suppresses non-equilibrium solution
ISW2B	Input control variable; selects types of input data for reservoir calculation
ISW3A	Input control variable; 0 suppresses equilibrium solution
ISW3B	Input control variable; 0 suppresses boundary layer calculations
ISW4A	Input control variable; must be nonzero if another case follows, 0 for last case
ISW4B	Input control variable; nonzero gives dump in boundary layer routine BLAYER
ISW5A	Input control variable; nonzero gives dump tracing the execution of subroutine RESTMP, which computes the reservoir temperature from mass flow and reservoir pressure or stagnation enthalpy
ISW5B	Input control variable, nonzero gives dump in subroutines EXACT, COMM, NONEQ and PRTA for debugging nonequilibrium calculations
ISW6A	Input control variable; if positive, only the reservoir equilibrium calculation is done; if negative, edit of species thermal properties is produced
ISW6B	Input control variable; 0 suppresses the output of species mole fractions in the free-stream and model-point output

IUPD	Indicator (initially 1), reset to 0 when switch from inverse method to direct integration is made in nonequilibrium solution
IZERØ	0 (set in READ)
JJK	Indicator (initially 0), reset to 1 in NEWRAP when electrons are dropped from the equilibrium calculation
LC	(Not used)
M1	Index equal to c+1
NFIT	Indicator. 0 if thermo fit data are not used for any species
NIT	Counter for number of iterations in Newton-Raphson procedure
NNN	Integration step counter
NNS	Number of times Δx has been increased by current value of SC
NQS	Number of successful integration steps before Δx is increased
NQT	(Not used)
NTEST	Number of iterations allowed in Newton-Raphson procedure
IGJ(J)	Indicator; 1 if thermo-fit data are used for J^{th} species, 0 if not
IGM(J)	Number of electronic levels for the J^{th} species
ITB(I)	Temporary storage for indices necessary to restart step in nonequilibrium calculation

KUR(I,J) U_{ij} , third body matrix
 LPIJ(I,J) α_{ij} , number of atoms of the j^{th} chemical
 element in a molecule of the i^{th} species
 ACØM(I) Hollerith description for case
 ELEMENT(I) Chemical symbols for elements (Hollerith)
 HP(I) Symbols for species (Hollerith)

4.2 Common /AEGEØM/

SQRTA Square root of the geometric area ratio
 S1 Geometric area ratio (reset to the effective
 area ratio in AESOLN)
 S2 Derivative of geometric area ratio (reset
 to derivative of the effective area ratio
 in AESOLN)

4.3 Common /AREA/

ATPI(J,I) Downstream boundary of the j^{th} section in
 the i^{th} profile (cm from the throat,
 positive downstream)*
 PARAM(L,J,I) For $L = 1$ to 3, the parameter values for the
 j^{th} section in the i^{th} profile. See Sec.
 4.3 of Volume I
 RTHCM(I) Throat radius (cm) for the i^{th} profile
 NSECT(I) Number of sections in the i^{th} profile
 NSECTU(I) Number of upstream sections in the i^{th} profile
 ISHAPE(J,I) Shape index for j^{th} section in the i^{th} profile

*"ATPI" is an acronym for "area transfer point input"

NPROFL(I) Index of i^{th} profile in the precoded data; NPROFL(1) is equivalent to NOZZLE.

NPRFLS Number of profiles (1 for a nozzle, 2 for a channel)

NBL Index (1 or 2) of the profile which diverges from the channel axis least rapidly downstream of the throat

4.4 Common /AVG/

WSAVE Parameter controlling the averaging distance for the boundary layer correlation parameter, n

4.5 Common /BL/

DELBL(L) Displacement thickness of the boundary layer on the L^{th} profile divided by R_0 . For two-dimensional and axisymmetric nozzles, only DELBL(1) is used.

BLINT(L) Integral I for the boundary layer on the L^{th} profile, eq. I(172)

XZERØ Position at which boundary layer is assumed to begin (negative value, cm from the throat)

TWALL Nozzle wall temperature ($^{\circ}\text{K}$)

CPWALL Specific heat of the gas at the wall temperature ($\text{cal/g } ^{\circ}\text{K}$)

VISRØT Ratio of viscosity in the reservoir to the reservoir temperature ($\text{poise}/^{\circ}\text{K}$)

DIAM(L) Throat diameter for the L^{th} profile (inches)

SW $S_w = (h_w/h_0) - 1$, where h_w is the gas enthalpy at the wall and h_0 the free-stream stagnation enthalpy

R0 Characteristic length used in boundary layer calculations (cm); equal to the throat radius for two-dimensional and axisymmetric nozzles; equal to $\sqrt{A_*'/\pi}$ for channels, where A_*' is the geometric cross sectional area of the channel at the throat

JDIM Dimensionality index, 0 for two-dimensional nozzles, 1 for axisymmetric nozzles

IPØINT Index of flow solution points at which boundary layer calculations are done

4.6 Common /BLNE/

XI(I) For $i = 1$ and 2 , the streamwise coordinate along the nozzle or channel surface for the i^{th} profile (cm)

ØRDINP(I) The quantity δ for the boundary layer on the i^{th} profile at the preceding flow point, eq. I(171)

DELBLP(I) $\delta^*/R0$ for the boundary layer on the i^{th} profile at the preceding flow point

DDBLP(I) $(d\delta^*/dx)/R0$ for the boundary layer on the i^{th} profile at the preceding flow point

AMPP Mach number at the point preceding the previous flow point

AMP Mach number at the preceding flow point

CØUPLD Logical variable, set to .TRUE. when the coupling of the boundary layer to the inviscid flow is switched on

ISMD The quantity $(30-i)$ in eq. I(418)

4.7 Common /BLOUT/

REPF	Reynolds number per foot (ft^{-1})
THETA(L)	Momentum thickness for the boundary layer on the L^{th} profile, divided by R_0
SN(L)	Correlation parameter n , eqs. I(158) and I(174), for the boundary layer on the L^{th} profile
XSN(L)	Averaged correlation parameter \bar{n} , eq. I(217), for the boundary layer on the L^{th} profile
PRREF	Prandtl number at the reference temperature
HR	Recovery enthalpy (cal/g)
QWDØT(L)	Heat flux to the wall surface with the L^{th} profile ($\text{Btu}/\text{ft}^2\text{sec}$)
TAUW(L)	Shear stress on the wall surface with the L^{th} profile ($\text{lb-f}/\text{ft}^2$)

4.8 Common /BLRAD/

YOZO	Product of the two throat radii for a channel (cm^2)
------	---

4.9 Common /CERRØR/

FAILED	Logical variable, set to .TRUE. when a convergence test on an integration step is failed in subroutine DERIVS or RNKT
--------	---

4.10 Common /CHAN/

CPl(I)	Specifications for standard channel no. 1: I = 1 Index of first channel profile in compiled-in list of profiles
--------	--

I = 2 Index of second channel profile in
compiled-in list of profiles

I = 3 Hollerith name for channel

I = 4 Index (1 or 2) of the profile which
diverges from the channel axis least
rapidly downstream of the throat

I = 5 Hollerith name for facility

CP2(I) Specifications for standard channel no. 2

CP3(I) Specifications for standard channel no. 3

CP4(I) Specifications for standard channel no. 4

CP5(I) Specifications for standard channel no. 5

4.11 Common /COLDSP/

CGMW Molecular weight of the cold gas mixture
(g/mole)

QPJ(J) Mole fraction of the Jth species in the cold
gas mixture

IJCS(J) Index of the Jth cold species in the list of
species for the current problem

NCS Number of cold species

4.12 Common /CONVRT/

CF(1) Conversion factor from nondimensional
temperature CT to temperature in °K

CF(2) Conversion factor from nondimensional
pressure PRES to pressure in atm

CF(3) Conversion factor from nondimensional
density RHØ to density in lbm/ft³

CF(4)	Conversion factor from nondimensional velocity SU to velocity in ft/sec
CF(5-7)	Conversion factors equal to 1
CF(8)	Conversion factor from viscosity in poise to viscosity in lbm/ft sec
CF(9)	Conversion factor equal to 1
CF(10)	Conversion factor from length in cm to length in inches
CF(11)	Conversion factor from nondimensional enthalpy CH to enthalpy in Btu/lbm
CF(12)	Conversion factor equal to 1
CF(13-16)	Conversion factors from nondimensional boundary layer thicknesses DELBL and THETA to thicknesses in inches
CF(17-26)	Conversion factors equal to 1

4.13 Common /DRV/

SHJDGJ $(R_0 T_0)^{-1} \sum H_j d\gamma_j/dx$, where H_j is the molar
enthalpy of the jth species and γ_j is the
concentration of the jth species in moles
per gram of mixture (set in subroutine EXACT)

4.14 Common /EELEM/

EEPl(J)	Data for first input-defined element
.	J = 1 Atomic number
.	J = 2 Atomic weight (g/mole)
.	
EEPl0(J)	Data for tenth input-defined element
	J = 1 Atomic number
	J = 2 Atomic weight (g/mole)

4.15 Common /ELEM/

EP1(I) Specifications for standard chemical element
no. 1:

EP1(1) Hollerith name

EP1(2) Atomic weight (g/mole)

EP2(I) Specifications for standard chemical element
no. 2

.

.

.

EP10(I) Specifications for standard chemical element
no. 10

4.16 Common /EQC/

This block is used by subroutine EQCALC to communicate
the results of its thermochemical equilibrium calculations
to the calling routine.

ZCAP(J) Mole fraction X_j of Jth species

ZSEN Entropy (cal/g °K)

ZCH Nondimensional molar enthalpy, $(R_0 T_0)^{-1} \sum x_j H_j$

ZCM Mean molecular weight

ZRBP Effective density ($\tilde{\rho}_0$) in reservoir for
imperfect-gas model

ZRHØ Nondimensional density, ρ/ρ_0

4.17 Common /EQC2/

ZPZ $\ln (p/p_0)$, logarithm of the nondimensional
pressure

ZGMU(I) $\ln X_i$, logarithm of the mole fraction for the
Ith chemical species

4.18 Common /ERROR/

ERR Logical flag set to "true" in DUMP routine; if "true", control is passed back to Main program, DUMPEX is called to print a dump of common data, and case is terminated

4.19 Common /GLIM2/

NØREAC(I) Logical control variables; when NØREAC(I) is set to .TRUE., the i^{th} reaction is suppressed

4.20 Common /INGNE/

AIN(K,I) The inverse A_{ki} of the square submatrix α_{ij} of the matrix α_{ij} , specifying the number of atoms of the j^{th} element per molecule of the i^{th} species; see eq. I(3)

4.21 Common /LN/

ISATØM Index of atom used in Lewis number calculation, in master list of species

ISMØL Index of molecule used in Lewis number calculation, in master list of species

JATØM Index of atom used in Lewis number calculation, in list of species for current problem

JMØL Index of molecule used in Lewis number calculation, in list of species for current problem

4.22 Common /MASSFL/

SMASS Sonic mass flux based on total mass flow and geometric throat area ($\text{g}/\text{cm}^2\text{sec}$)

CTMXXI Temperature ($^{\circ}\text{K}$) above which species thermal properties are computed from the thermo fit for those species for which thermo fits are supplied

TSTØPI Free-stream temperature at which the flow solutions will be terminated (⁰K)

IS(J) Index of the Jth species in the current problem, in the master list of species

4.23 Common /MIXT/

GP1(I) Specifications of standard gas mixture no. 1:

- | | |
|------------|--|
| I = 1 | Hollerith name of mixture |
| I = 2 | Number of chemical elements in mixture, including e ⁻ if the gas model contains ion species |
| I = 3 | Number of chemical species in mixture, including e ⁻ if model contains ion species |
| I = 4 | Number of reactions included in gas model |
| I = 5 | Number of ions in gas model, excluding e ⁻ |
| I = 6-15 | Indices of elements present in mixture, in master list of elements; if electrons are present, they are the first element |
| I = 16-25 | Mole fractions (QPJ) of the cold species |
| I = 26-45 | Indices of the species included in the gas model, in the master list of species |
| I = 46-109 | Indices of the reactions included in the model, in the master list of reactions |

I = 110-119	Indices of the cold species, in the master list of species
I = 120	Number of cold species
I = 121	Atom index for Lewis number calculations, in the master list of species
I = 122	Molecule index for Lewis number calculations, in the master list of species
I = 123	INT. If INT = 0, electron temperature equals gas temperature. If INT > 0, the model includes electronic nonequilibrium, and INT is the index of the extra reaction properties required in TNEP(I,INT)
I = 124	Indicator for inclusion (1) or exclusion (2) of Fay-Riddell Lewis number factor in the stagnation-point heat flux
GP2(I)	Specifications of standard gas mixture no. 2
.	.
.	.
.	.
GP6(I)	Specifications of standard gas mixture no. 6
4.24 <u>Common /MODPAR/</u>	
XMP1	Initial distance from throat at which model condition calculations are to be done (cm)
DXMP	(Not used)
FSTAG	Control variable; 0. value suppresses frozen shock calculations at model points, negative value suppresses equilibrium shock

CATFAC	Catalytic efficiency parameter $\bar{\alpha}$ for stagnation point heat flux calculations; 0 for noncatalytic wall
TMØDEL	Model wall temperature for stagnation point heat flux calculations ($^{\circ}\text{K}$)
XMØDPI	Initial distance from the throat at which model condition calculations are to be done (inches)
DXMØDP	(Not used)
TPLATE	Wall temperature for calculations of heat flux to a flat plate at zero angle of attack ($^{\circ}\text{K}$)
KDIM	Control variable for stagnation point heat flux calculations; 0 for two-dimensional model geometry, 1 for axisymmetric model

4.25 Common /MØDPT/

TSDIAM(I)	Test-section diameters at which model condition calculations are to be done (inches); in channel flow problems, specified channel widths at which flow calculations are to be done
TSAR(I)	Geometric area ratios at the nozzle stations defined by TSDIAM(I)
NTS	Number of specified test section diameters
MBL	In channel flow problems, index of the profile defining the channel width

4.26 Common /MØDP2/

MØDLPT	Logical indicator, .TRUE. when the current step in the nonequilibrium solution is at a model point
--------	--

4.27 Common /NEQ/

ØMDST(L) 1 - δ^*/R_0 for the Lth profile at the sonic point

DDELBL(L) $(d\delta^*/dx)/R_0$, derivative of nondimensional displacement thickness for the boundary layer on the Lth profile

4.28 Common /NEWMP/

FACMP Factor by which distance from throat is increased in a geometric sequence of model points

NMØDPT Number of model points in geometric sequence

4.29 Common /NØZZ/

ZP1(i) Specifications of standard nozzle profile no. 1:

I = 1 Throat radius in cm

I = 2 Distance upstream of the throat of the point at which the boundary layer is assumed to originate (negative value, cm)

I = 3 Number of profile sections upstream of the throat

I = 4 Number of profile sections downstream of the throat

I = 5-16 Shape indices (ISHAPE) of the profile sections (in order from upstream to downstream):

ISHAPE = 1 Straight section

ISHAPE = 2 Circular section convex toward axis

ISHAPE = 3 Circular section concave
toward axis

I = 17-27 Downstream boundaries of profile
sections (distances from the throat,
positive downstream, cm)

I = 28-63 Parameters defining the profile
sections (lengths in cm). * There
are three parameter values P_1 , P_2 ,
 P_3 for each profile. For ISHAPE =
1, the equation of a straight
profile is

$$r(x) = P_1 + P_2x$$

For ISHAPE = 2 or 3

P_1 = distance of circle center
from axis

P_2 = x-coordinate of circle center

P_3 = circle radius

I = 64 Hollerith facility name

ZP2(I) Specifications of standard profile no. 2

·
·
·

ZP20(I) Specifications of standard profile no. 20

4.30 Common /ØUIMØD/

TB7 $N_0 \rho_0$, where N_0 is Avogadro's number (6.0225×10^{23} particles per mole) and ρ_0 is the
density in the reservoir in g/cm^3 .

*See Section 4.7 of Volume II

4.31 Common /ØUTPUT/

FVØUT(I) Array used for output of flow variables; set
in subroutine ØUT1 (entry ØUT2)

- I = 1 Axial coordinate, x (inches)
- I = 2 Temperature ($^{\circ}\text{K}$)
- I = 3 Static enthalpy (Btu/lb)
- I = 4 Pressure (atm)
- I = 5 Density (lb/ft^3)
- I = 6 Nozzle diameter (inches); geometric
 area ratio in channel solutions
- I = 7 Velocity (ft/sec)
- I = 8 Mach number
- I = 9 Entropy (Btu/lb $^{\circ}\text{R}$)
- I = 10 Frozen specific heat ratio
- I = 11 Effective area ratio
- I = 12 Reynolds number (per ft)
- I = 13 Molecular weight (g/mole)
- I = 14 Viscosity ($\text{lb}/\text{ft sec}$)
- I = 15 Electrical conductivity (mho/cm)
- I = 16 Geometric area ratio; width of face
 with first profile in channel
 solutions
- I = 17 Displacement thickness (inches)
- I = 18 Momentum thickness (inches)

- I = 19 Heat flux to wall (Btu/ft²sec)
- I = 20 Shear stress on wall (lbf/ft²)
- I = 21 Recovery enthalpy (Btu/lb)
- I = 22 Prandtl number at the reference temperature
- I = 23 Stanton number
- I = 24 Reynolds number based on θ
- I = 25 Re_{θ} for boundary layer transition
- I = 26 Width of face with second profile (channel solutions only)
- I = 27 Displacement thickness on second profile (inches)
- I = 28 Momentum thickness on second profile (inches)
- I = 29 Heat flux to wall on second profile (Btu/ft²sec)
- I = 30 Shear stress on wall on second profile (lbf/ft²)
- I = 31 Recovery enthalpy (Btu/lb)
- I = 32 Prandtl number at the reference temperature
- I = 33 Stanton number on second profile
- I = 34 Re_{θ} on second profile
- I = 35 Re_{θ} for transition

GJMF(J)

(Not used)

4.32 Common /POLYAT/

THEVE(K,J) For K = 1-4, the four characteristic vibrational temperatures of the Jth species in the current gas model, if this species is a linear triatomic molecule. In subroutine READ, the entries K = 2-4 are set to values in $^{\circ}\text{K}$. In subroutine INIT, these values are nondimensionalized by dividing them by the reservoir temperature, and the value K = 1 is set

4.33 Common /RDLIST/

IGAS Index of standard gas mixture

IGASØ IGAS value in preceding case (if any)

NØZZØ Nozzle profile index for preceding case

ICHANØ Channel index for preceding case

4.34 Common /RDMAIN/

HS Input value of stagnation enthalpy (cal/g)

SUPGØ Logical control for suppressing output of the gas model data when the model is the same as in the preceding case

MFITER Input control for the iteration to take the displacement thickness into account in the reservoir condition calculations; zero value suppresses the iteration

NØ^{'''}RAN Input logical control for suppressing all transport property calculations

4.35 Common /RDMØD/

LEWIS Index controlling use of the Fay-Riddell Lewis number factor in stagnation-point heat flux calculations. For LEWIS = 1, the factor is used; for LEWIS = 2, it is omitted

IAMBIP Index specifying whether the diffusion process upon which the Lewis number is based is ambipolar (IAMBIP = 2) or not (IAMBIP = 1)

4.36 Common /RDØUT/

FLØW Total mass flow in lbm/sec

FACNAM Hollerith facility name

CHANAM Hollerith channel name

LIMØUT Index, nonzero if the boundary layer parameters N and XSN are to be printed in the output, equal to 0 if not

4.37 Common /RDTR/

ISW8B Index controlling diagnostic dumps in the transport property routines. If ISW8B = 0, these dumps are omitted. If ISW8B > 0, the PUTQIN dump is produced once every ISW8B times the subroutine PUTQIN is called. If ISW8B < 0, the PUTQIN dump is suppressed.

4.38 Common /RDWEDG/

ANGLE(I) Angles of attack (degrees)

RADLE(J) Radii of leading edge (inches)

WX1 Initial distance from leading edge of wedge (inches)

DWX Increment in distance from leading edge (inches)

WXI(K)	Specified distances from leading edge (inches)
TWEDGE	Surface temperature of wedge model ($^{\circ}\text{K}$)
WK	Leading-edge drag coefficient, k
NWX	Number of distances from leading edge (at increment of DWX) at which surface conditions on wedge model are to be calculated
NANGLE	Number of angles of attack
NRADLE	Number of leading-edge radii
WEDGEM	Logical control variable; if "false", conditions on wedge are not calculated
AXISYM	Logical control variable; if "false", stagnation point heat fluxes are not calculated
ISW9B	Control variable, normally 0, with the following possible optional values: <u>±</u> 1 Print shock ordinate Y_s <u>±</u> 2 Print nondimensional coordinate ζ <u>±</u> 3 Print both Y_s and ζ If ISW9B is negative, calculations are done using the unmodified Cheng-Kemp theory as well as the modified theory

4.39 Common /REAC/

RP1(I)	Specifications of standard reaction no. 1:
I = 1	Coefficient A in formula for forward reaction rate constant ($\text{cm}^3/\text{mole sec}$ or $\text{cm}^6/\text{mole}^2 \text{ sec}$)
I = 2	Exponent η in formula for rate constant

I = 3	Activation energy E_a in formula for rate constant (cal/mole)
I = 4	Equal to 1.0 if a third-body list is provided for the reaction, equal to 0.0 if not
I = 5	Number of reactant species (≤ 3)
I = 6	Number of product species (≤ 3)
I = 7-9	Indices of reactant species in master list of species
I = 10-12	Indices of product species in master list of species
I = 13-15	Numbers of molecules of reactants
I = 16-18	Numbers of molecules of products
I = 19	Number of third bodies
I = 20-29	Indices of third-body species in master list of species

RP2(I)	Specifications of standard reaction no. 2
.	.
.	.
.	.
RP92(I)	Specifications of standard reaction no. 92

4.40 Common /READAT/

PRESAI	Reservoir pressure input (atm)
DELT1I	Input value of nondimensional temperature decrement used in the frozen and equilibrium flow calculations and in starting the non-equilibrium solution

TPRNTI	The free-stream nonequilibrium solution is printed out at temperature intervals greater than or equal to TPRNTI times the reservoir temperature. For TPRNTI = 0., every step is printed
DELTXI	Input value of initial step size in x for nonequilibrium integration (cm)
CXMAXI	Distance beyond the throat at which the flow solutions are stopped (inches)
CTAPI	Reservoir temperature input ($^{\circ}$ K)
XZERØI	Nozzle or channel inlet position at which the boundary layer is assumed to originate (negative value, measured in inches upstream from the throat)
PARAMI (L,M,N)	Input array of parameter values for nozzle or channel profiles (see definitions of code inputs, Group 4, in Section 2.3 of Volume II)
BZERØI	Input of constant in imperfect gas correction
HSTAG	Input stagnation enthalpy (Btu/lb)
READXS	Input control for reading cross section data for transport property calculations
READG	Input control for reading data on elements, species, and/or reactions
AAMS	Input control for automatic air model selection
AXIMØD	Input control for suppressing stagnation point model condition calculations if only wedge calculations are desired
ICASE	Case number in the current job
NEELS	Number of elements being defined in the input

ISW7B	Input control for suppressing the output of the boundary layer parameters N and XSN
INT	Indicator for electronic nonequilibrium in a gas model
ICHAN	Index of standard channels
NQSI	Number of successful integration steps required before the step size in the non-equilibrium calculation is increased
INEQVI	Input control for equilibrium or frozen molecular vibration
NRECO	Input of number of records already on data tape at beginning of run
JCS(I)	Indices of cold species in master list of species
ISCI	Input of the number of chemical elements in the gas model
ISSI	Input of the number of chemical species in the gas model
ISRI	Input of the number of reactions in the gas model
ICI	Input of the number of ions in the gas model
IE(I)	Input indices of elements present in the gas model, in the master list of elements
IR(I)	Input indices of reactions in the gas model, in the master list of reactions
NSECTS(I,J)	For I = 1, number of upstream sections in the curvefit for the Jth profile; for I = 2, number of downstream sections
IEEP(I)	Indices assigned to the input-defined elements in the master list of elements

4.41 Common /RENE/

GAMIN	Concentration (moles/g) below which a species will be frozen if it decreases so rapidly that it controls the integration step size.
HTEST	Maximum relative change in total enthalpy in an integration step.
TETEST	Maximum relative change in electron temperature in an integration step
QTEST	Criterion value for maximum allowable change in energy transfer to electron gas in an integration step
DCHLL	Parameter used in selecting initial integration step size; the initial Δx is not allowed to exceed $0.01 \delta \chi_i _{\min} / DCHLL$.
DCHRAT	Parameter controlling the artificial increase in reaction rates in the perturbation solution to avoid premature startup of the numerical integration (Section 7.3.7 of Volume I)
CCHI	Parameter C_χ in criterion for switch from perturbation solution to numerical integration
TTEST	Maximum relative change in the gas temperature in an integration step
GTEST	Maximum relative change in a species concentration in an integration step.

4.42 Common /RESRP/

VISCR	Viscosity in the reservoir (poise)
PRR	Prandtl number in the reservoir
SIGR	Electrical conductivity in the reservoir (mho/cm)

FLEWR Lewis number in the reservoir

4.43 Common /SPEC/

SPl(I) Specifications for standard species no. 1:

- | | |
|-----------|---|
| I = 1 | Hollerith name for species |
| I = 2 | Number of elements in species (≤ 3) |
| I = 3-5 | Indices of elements in the master list of elements |
| I = 6-8 | Numbers of atoms of elements |
| I = 9-14 | Thermo-fit parameters a, b, c, d, e, k |
| I = 15 | Formation enthalpy (cal/mole) |
| I = 16 | Number of atoms in a molecule of the species |
| I = 17 | Chemical constant, b |
| I = 18 | Characteristic vibrational temperature ($^{\circ}\text{K}$) |
| I = 19 | Number of electronic levels (≤ 10) |
| I = 20 | 1 if thermo-fit is used, 0 if not |
| I = 21-30 | Degeneracies of electronic levels |
| I = 31-40 | Energies of electronic levels (cal/mole above ground state) |
| I = 41-43 | Second, third, and fourth characteristic vibrational temperatures for linear triatomic species ($^{\circ}\text{K}$) |

4.44 Common /SS/

- CAS Square of the sound speed divided by the non-dimensional temperature T/T_0 (cm^2/sec^2)
- US Standard velocity used for nondimensionalizing the flow velocity, $\sqrt{R_0 T_0 / W_0}$, where T_0 is the reservoir temperature, W_0 the reservoir molecular weight, and $R_0 = 8.31436 \times 10^7$ erg/mole $^\circ\text{K}$

4.45 Common /STAG/

- T2E Initial estimate of temperature behind equilibrium normal shock ($^\circ\text{K}$)
- EPSLØN Initial estimate of density ratio across equilibrium normal shock
- T2F Initial estimate of temperature behind frozen normal shock ($^\circ\text{K}$)
- EPSF Initial estimate of density ratio across frozen normal shock

4.46 Common /STEPS/

- NSTEPS Number of successful integration steps since the last printout of conditions at a flow point

4.47 Common /SWITCH/

- XPB The corrected x-coordinate of the flow point preceding the point at which BLAYER is first called with x corrected for the displacement thickness in the frozen and equilibrium solutions (cm)
- DXPB The difference in corrected x values between the point at XPB and the preceding point

(1) W Constant parameter w determining the characteristic distance over which the boundary layer correlation parameter n is averaged; see eqs. I(218) and I(217)

4.48 Common /TAPØUT/

XXX(I) Block of storage used to set up data which are to be saved on binary tape for subsequent processing by other programs

ITPØUT File number of binary output tape

NRCØUT Number of records written on binary tape during current case of job

IFLØW (Not used)

ITYPER Type of record written on binary tape:

- (1)
- 1 "Case" record with overall definition of flow solution being computed
 - 2 "Point" record containing free-stream flow variables
 - 3 "Model" record containing data for stagnation point model

IMP Index of model points written onto binary tape in current solution

DATAPE Logical control variable; if "false", no data are written onto binary tape

4.49 Common /TEMPRY/

SAVEC(J) Species mole fractions for which transport property calculations are done

4.50 Common /THRT/

(1)

RSA Factor by which cross sectional area of flow has been rescaled in nonequilibrium solution

4.51 Common /TNCE/

SUMGH $\sum_{j=1}^n \gamma_j$, sum of the molar concentrations for all species except the electron (mole/g)

SCPGH $\sum_{j=1}^n C_{pj} \gamma_j / R_0$, sum of the products of the molar heat capacities and molar concentrations for all species except the electron, divided by the universal gas constant

QDPR Power radiated from the gas, cal/cm³-sec

QDPE Net power transferred to the electron gas (cal/cm³-sec)

4.52 Common /TNDC/

XMJATD(J,L) Chemical potential at standard pressure (μ_j^0) for the Jth species, evaluated at the Lth temperature (L = 1, heavy particle temperature; L = 2, electron temperature), divided by R_0 times the Lth temperature

CLNTD(L) Natural logarithm of the Lth temperature (L = 1, heavy particle temperature; L = 2, electron temperature)

CTD(2) Nondimensional temperatures (L = 1, T/T_0 ; L = 2, T_e/T_0)

4.53 Common /TNE/

TN1(I) Specifications of data for standard electronic nonequilibrium model no. 1 (helium):

I = 1-25 KTF(I), indicator for type of formula for forward rate constant of the Ith reaction; see Sec. 4.4 of Vol. II

I = 26-50 KTR indicators for the reverse reactions in the gas model. KTR = 0 if the backward rate constant k_r is zero; KTR = 1 if $k_r = k_r(T)$; and KTR = 2 if $k_r = k_r(T_e)$

I = 51-75 ITR values for the reactions in the gas model. ITR is an indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses. The significance of its values is discussed in Sec. 4.4 of Vol. II

I = 76-100 ϵ_0 parameters for reactions

I = 101-125 Values of the parameter "a" for reactions with ITR = 1

I = 126-155 Temperature values for table of elastic collision cross section, $\bar{Q}(1,1)$

I = 156-185 $\bar{Q}(1,1)$ values for table

I = 186 Parameter b for reactions with KTF = 4

TN2(I) Specifications of data for electronic non-equilibrium model no. 2 (argon)

4.54 Common /TNEQ/

TLIST(1) For I = 1-30, temperatures for elastic collision cross section table for current gas model ($^{\circ}\text{K}$)

PØM(I) For I = 1-30, cross section values for elastic collision cross section table for current gas model

4.55 Common /TNERK/

SDTE	$\Delta T_e/T_0$, change in electron temperature in integration step
CTEB	T_e/T_0 at the start of the integration step
DCHA	$(W_0/R_0T_0) (dh_0/dx)$, rate of change of non-dimensional total enthalpy
CHB	W_0h_0/R_0T_0 , nondimensional total enthalpy at start of integration step
SDCHA	$\Delta (W_0h_0/R_0T_0)$, change in nondimensional total enthalpy in integration step
DQMAX	D_{gm} , criterion value for convergence test on changes in energy transfer to the electron gas
IFAIL	Indicator for cause of integration step failure:

<u>IFAIL</u>	<u>Meaning</u>
0	Failure in COMM or DERIVS (called from RNKT)
1	Test on T in NØNEQ
2	Test on T_e in NØNEQ
3	Test on h_0 in NØNEQ
4	Test on DL_0GA in NØNEQ
5	Test on QDPE in NØNEQ
11-30	Test on GJ(IFAIL-10) from RNKT in NØNEQ
31-50	Test on GJ(IFAIL-30) from element conservation in NØNEQ

<u>IFAIL</u>	<u>Meaning</u>
60	Failure in DERIVS or CØMM (called from NØNEQ)
-1 to -ISS	Test on sign of GJ(-IFAIL) in RNKT (GJ2 or GJ3)
-(ISS+1)	Test on sign of T in RNKT (GJ2 or GJ3)
-(ISS+2)	Test on sign of T_e in RNKT (GJ2 or GJ3)
-(ISS+3)	Test on sign of h_0 in RNKT (GJ2 or GJ3)
-(ISS+4)	Test on QDPE in RNKT (GJ2 or GJ3)
-31 to (-30+ISS)	Test on sign of GJ(-IFAIL-30) in RNKT (GJ4, Runge-Kutta)
-(31+ISS)	Test on sign of T in RNKT (GJ4, Runge-Kutta)
-(32+ISS)	Test on sign of T_e in RNKT (GJ4, Runge-Kutta)
-(33+ISS)	Test on sign of h_0 in RNKT (GJ4, Runge-Kutta)
-(34+ISS)	Test on QDPE in RNKT (GJ4, Runge-Kutta)
-61 to -(60+ISS)	Test on sign of GJ(-IFAIL-60) in RNKT (GJ4, Treanor)
-(61+ISS)	Test on sign of T in RNKT (GJ4, Treanor)
-(62+ISS)	Test on sign of T_e in RNKT (GJ4, Treanor)

IFAILMeaning

-(63+ISS) Test on sign of h_0 in RNKT
(GJ4, Treanor)

-(64+ISS) Test on QDPE in RNKT (GJ4,
Treanor)

4.56 Common /TNONEQ/

CTE	Electron temperature divided by gas temperature in reservoir
DTE	Derivative of CTE with respect to x , i.e., $d(T_e/T_0)/dx$
BPAR	Parameter b for reactions with $KTF = 4$
EPAR(I,JR)	EPAR(1,JR) = parameter ϵ_0 for the JRth reaction in cal per N_0 reactions; EPAR(2,JR) = parameter a for the JRth reaction if ITR(JR) = 1
NT	Indicator for electronic nonequilibrium in the gas model; for $NT = 1$, the gas and electron temperatures are assumed to be equal; for $NT = 2$, they are not
ITR(JR)	Indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses in the JRth reaction of the current gas model. See discussion in Section 4.4 of Vol. II
KTF(JR)	Indicator for temperature dependence of the forward rate constant k_f for the JRth reaction in the current gas model; see Sec. 4.4 of Vol. II

KTR(JR) Indicator for temperature dependence of the backward rate constant k_r for the JRth reaction in the current gas model:

KTR = 0	$k_r = 0$
KTR = 1	$k_r = k_r(T)$
KTR = 2	$k_r = k_r(T_e)$

ICH(J) Indicator for charged species; equal to 1 if the Jth species is neutral, equal to 2 if the species is an ion

IPA(I) Species index for the product atom in the Ith reaction; if KTF(I) is not equal to 4, IPA(I) is set to zero

4.57 Common /TRANI/

III(K) For $K = 1$ to a maximum of 5, the index of the first species in the Kth pair of species to which the cross section calculated in the i^{th} step of the transport property calculation is applied. The common block provides storage for $i = 1$ to 100

4.58 Common /TRANJ/

JJi(K) For $K = 1$ to a maximum of 5, the index of the second species in the Kth pair of species to which the cross section calculated in the i^{th} step of the transport property calculation is applied. The common block provides storage for $i = 1$ to 100. Only pairs with $III(K) \leq JJi(K)$ are used

4.59 Common /TRANS1/

T Temperature in $^{\circ}K$ for transport calculations

Q(K,I,J)

Q(K,I,J) is used in the cross section computations to store the averaged cross sections for the species pair (I,J) as follows:

$$Q(1,I,J) = \bar{\Omega}_{ij}(1,1),$$

$$Q(2,I,J) = \bar{\Omega}_{ij}(2,2),$$

$$Q(3,I,J) = B_{ij}^* \bar{\Omega}_{ij}(1,1).$$

At other stages in the calculations, Q(K,I,J) is also used in subroutine TRANSP to store the matrix elements for the transport calculations, and in subroutine XSECT to indicate the species pairs for which cross section data have been supplied

ZM2(I)

Quantity proportional to molecular weight of species I, eqs. (171)

4.60 Common /TRANS2/

KKQ(L)

Index of option to be used for step L of cross section computations in input or precoded data; see Section 4.4 of Vol. II

NNQ(L)

Number of species pairs in IIQ, JJQ arrays for which option KKQ(L) is specified in input data

ISEQ(K)

Sequencing array specifying the order in which the defined steps of the transport cross section calculation are to be carried out

NNKQ

Number of steps specified for cross section computations

I(J)

Temporary array of species indices used by subroutine XSECT in preparing edited cross section data

4.61 Common /TRANS3/

B(I,L) Intermediate variable in transport calculations;
see KINT and KANDMU, Sections 3.27 and 3.28

BR(I,L) Reciprocal of B(I,L)

A(L) Intermediate variable in transport calculations;
see KANDMU, Section 3.27

X(I) Mole fraction of species I in gas mixture

DH(I) Nondimensional internal specific heat
($W_i C_{pi} / N_0 k - 5/2$) for species I in gas mixture

4.62 Common /TRANS4/

TL(I) Independent variable (temperature) for tabulated
cross section data

ØMEGAL(I) Value of first parameter determining cross
sections at TL(I) for tabulated cross section
data

ASTAR(I) Value of second parameter determining cross
sections at TL(I) for tabulated cross section
data

BSTAR(I) Value of third parameter determining cross
sections at TL(I) for tabulated cross section
data

4.63 Common /TRANS5/

N Number of species included in transport
calculations

IELEC IELEC=0 indicates that electrons are not
included in the calculations

ID1 Index in the current gas model of first
species used in Lewis number computation

ID2 Index in the current gas model of second species used in Lewis number computation

4.64 Common /TRANS7/

V(M) List of parameters for cross section computations in edited data

KQ(L) Option to be used for step L of cross section computations in edited data

NQ(L) Index of last species pair in IQ,JQ arrays to which step L of the edited cross section computations is to be applied

IQ(N) List of first indices for species pairs to which edited cross section data are to be applied

JQ(N) List of second indices for species pairs to which edited cross section data are to be applied

NKQ Number of steps in cross section computations for edited cross section data

4.65 Common /TRANS8/

NV(K) Number of parameters V required for option K of the cross section computations

N Number of species included in cross section computations

4.66 Common /TRANV/

VVi(K) For K = 1 to a maximum of 5, the input parameters for the ith step of the cross section calculations. The common block provides storage for i = 1 to 100

4.67 Common /TRPRØP/

VISC	Viscosity of mixture in poise
PRF	Frozen Prandtl number for mixture (dimensionless)
SIGMA	Electrical conductivity of mixture in mhos/cm
FLEWIS	Atom-molecule Lewis number of mixture (dimensionless)

4.68 Common /WMØD/

TSF	Frozen stagnation temperature (°K)
-----	------------------------------------

4.69 Main Program

ATEST	Previous value of A in the Newton-Raphson iteration to determine the parameter α in the density-area relation I(383)
ATH	Effective cross-sectional area of the flow at the throat, with allowance for the boundary layer displacement thickness (cm ²)
CHTH	Nondimensional enthalpy hW_0/R_0T_0 at the throat
CMTH	Molecular weight of gas mixture at the throat
D	$(\rho_*/\rho_0)^A (A+2) - 2$, where ρ_* is the density at the sonic point in the equilibrium flow solution and A is the current estimate of the parameter α in the density-area relation
DELT	Initial decrement of the nondimensional temperature for starting the perturbation calculation of the nonequilibrium flow
D1	$(\rho_*/\rho_0)^A [(A+2) \ln (\rho_*/\rho_0) + 1]$
ELTMIN	Case execution time in minutes

ET	Elapsed time from beginning of execution (seconds)
ETO	Elapsed time from beginning of execution at the start of a case (seconds)
GJ(I)	Species concentrations in moles/g
GJA(I)	Species concentrations in the upstream reservoir, moles/g
GMF(J)	Mole fraction of the J th species under equilibrium sonic conditions
HOP	Reservoir enthalpy (cal/g)
I	DØ index; also, index, in the list of species for the current problem, of the K th cold species
IRCØ	Indicator; value 1 indicates that call to TRANSX was skipped in the previous case because tables of species thermal properties were generated
ISØLN	Indicator for type of flow solution: ISØLN = 1 Frozen ISØLN = 2 Equilibrium ISØLN = 3 Nonequilibrium
IS3	Storage for input value of ISW3B; used to restore value after a case in which ISW3B was set to zero because NØTRAN = .TRUE.
K	Cold species index
N	Iteration counter in Newton-Raphson solution for parameter α in area-density relation
NCASE	Counter for cases in a run
NCØMPL	Number of cases successfully completed in a run

NERRØR	Number of failed cases in a run
NØZZLE	Nozzle profile index
NRCTØT	Total number of records written on tape 8 during a run
ØNE	1.0
PIV	3.14159
SUM1	$\ln (\rho_*/\rho_0)$
SUM2	A+2
SUM3	$(\rho_*/\rho_0)^A$
UTH	Nondimensional flow velocity (SU) at throat
VISCC	Viscosity in the reservoir (lbm/ft-sec)

4.70 Subroutine AESØLN

DELP(L)	Displacement thickness in cm, on Lth profile
DELSTP(L)	Displacement thickness at the throat in cm, on Lth profile
DENØM	One-fourth of the effective flow area at the throat for a channel (cm ²)
DYDX	Slope dy/dx for the first profile in a channel
DZDX	Slope dz/dx for the second profile in a channel
FAC1	Effective ordinate of first profile in a channel (cm)
FAC2	Effective ordinate of second profile in a channel (cm)

SQRTAE	Square root of the effective area ratio, $\sqrt{A_e}$
X	Axial coordinate in nozzle (cm)
Y	Geometric ordinate of first profile in a channel (cm)
Z	Geometric ordinate of second profile in a channel (cm)

4.71 Subroutine AGSØLN

A	Trial value of geometric area ratio
AE	Effective area ratio
AEC	Constant term in equation relating the geometric and effective area ratios for a channel
AG	Final solution for the geometric area ratio
DA	Dummy argument in call to GEØMAR; represents X-derivative of geometric area ratio (not used)
DEL(L)	Displacement thickness divided by RØ, for boundary layer on Lth profile
DELP(L)	Displacement thickness (cm) for boundary layer on Lth profile
DSTP(L)	Displacement thickness at the throat (cm) for boundary layer on Lth profile.
F	Function which is 0 when the correct geometric area ratio has been found
FØ	F value in previous step of iteration
ICØUNT	Iteration counter
L	DØ index over channel profiles

RNAME	Routine name for call to DUMP routine
SQRTA	Square root of geometric area ratio
UPDOWN	Indicator; 1.0 if downstream solution is desired by calling routine, -1.0 if upstream solution is desired
X	Axial coordinate in nozzle (cm)
XL	Temporary storage for previous value of X
XØ	Value of X in previous iteration
Y	Geometric ordinate of first profile in a channel (cm)
YOZO	$y_0 z_0$, product of profile ordinates at the throat; one-fourth of the geometric throat area in a channel (cm ²)
Z	Geometric ordinate of second profile in a channel (cm)

4.72 Subroutine AXFIT

AG	Geometric area ratio
UPDOWN	Indicator; -1.0 upstream of throat, 1.0 downstream

4.73 Subroutine AXSECT

I(J)	List of species for which cross section data are wanted (common array)
II	Index of first species in species pair
IV	Location of species II in the I array
JJ	Index of second species in species pair
JV	Location of species JJ in the I array

L	Running index for DØ loops
LL	Step in cross section computations where data are to be added
LQ	Position of species pair in IQ,JQ list
LQ1	Position in IQ,JQ list where additional species pairs are to be added
N	Number of species pairs added to IQ,JQ list
NN	Number of species from I array to be considered in adding data

4.74 Subroutine BLAYER

AE	Effective area ratio
AG	Geometric area ratio
AGJ	Factor proportional to r^{2j} in numerical integration from x_0 to the first flow point
AG1	Geometric area ratio in numerical integration from x_0 to the first flow point
AJ	Factor proportional to r^{2j}
ALPR	Natural logarithm of the Prandtl number at the reference temperature
AM	Mach number
AM1	Mach number in numerical integration from x_0 to the first flow point
AS	Sound speed (cm/sec)
AVCON	W/R0 parameter used in averaging the correlation parameter

AX	Intermediate quality, eq. I(194d) in curvefit to shear parameter
B	Coefficient of n in linear approximation to $N(n, S_w)$, eq. I(164)
BLINTI	Value of integral I , eq. I(172), at the beginning of a step in the boundary layer solution
BM1	B-1
BX	Intermediate quantity, eq. I(194e), in curvefit to shear parameter
CC	MA_g in low Mach number region of integration from x_0 to first flow point
CØSB(L)	Cosine of angle between tangent to L th profile and the nozzle axis
C1	Intermediate quantity in quadratic solution for dM/dx
C2	Intermediate quantity in quadratic solution for dM/dx
DADX	dA_g/dx , derivative of geometric area ratio
DLM	$d \ln M/dx$, logarithmic derivative of Mach number
DMDX	dM/dx , derivative of Mach number
DUDX	Derivative of nondimensional velocity
DX	Axial distance from the previous flow point to the current one
DXI	Change in ξ in the current step

DXØ	Size of last previous "final" step, apart from steps at model points
DXP	Distance from the point before the previous flow point to the previous flow point
DXSQ	$(DX)^2$
DX2	$DX + DXP$
DYZDX(L)	Derivative of the radius of the Lth profile with respect to the axial coordinate
EK1	Coefficient A in eq. I(164)
ENPRIM	n' , eq. I(170)
FINAL	Logical flag; "false" when BLAYER is called during an intermediate calculation of the non-equilibrium integration; "true" for the final calculation of conditions at each flow point
F0	Intermediate quantity in curvefit to shear parameter, eq. I(194)
F2	Intermediate quantity in curvefit to shear parameter
F3	Intermediate quantity in curvefit to shear parameter
F4	Intermediate quantity in curvefit to shear parameter
GAMMAE	Effective specific heat ratio, $\gamma_e = a^2 W/RT$, used in calculation of effective hypersonic parameter
GJ(I)	Species concentrations (moles/g)
H	Nondimensional enthalpy, hW_0/R_0T_0

HCØN	$R_0 T_0 / W_0$, factor for converting nondimensional enthalpy into cal/g
HE	Free-stream specific enthalpy (cal/g)
HF	Form factor, \hat{c}^* / θ
HFPl	HF + 1
HREF	Reference enthalpy
HTR	Incompressible form factor, H_{inc}
HTRPl	HTR + 1
HW	Enthalpy of the gas at the wall temperature
HO	Specific stagnation enthalpy (cal/g)
I	DØ index
ITYPE	Index for type of nozzle geometry ITYPE = 1 Two-dimensional nozzle ITYPE = 2 Axisymmetric nozzle ITYPE = 3 Rectangular channel
J	DØ index
K	DØ index over the cold species
L	DØ index running over the nozzle or channel profiles
LPR	Index, equal to LPRIME(L)
LPRIME(L)	Equal to 2 for L = 1, equal to 1 for L = 2
ØLDMF	Logical flag; "true" if no mole fraction has changed by 0.01 since the last flow point at which transport properties were calculated

ØMEGA	ω , viscosity-temperature exponent in eq. I (191)
ØM1	Max (1- ω , 0)
ØNE	1.
ØRDIN(L)	$\bar{\Phi}$ for the boundary layer on the Lth profile, eq. I(171)
ØRD1	A value of $\bar{\Phi}$ used in the numerical integration from x_0 to the first flow point
ØRD2	See ØRD1
P	Nondimensional pressure, p/p_0
PRPWR	(PRREF) ^{0.56}
PRW	Prandtl number at the wall temperature
R	Ratio of local nozzle radius to throat radius for axisymmetric nozzles
RA	Intermediate quantity in curvefit to Reynolds analogy factor
RAF	Reynolds analogy factor R_A in eq. I(195)
RB	Intermediate quantity in curvefit to Reynolds analogy factor
RER	Reynolds number based on R_0
RH	Nondimensional density
RMR	$\rho\mu$ ratio, $\rho_w\mu_w/\rho_e\mu_e$
RMMW	ρ_w ;
R1	Radius of a profile in the numerical integration from x_0 to the first flow point

R2	See R1
SIGH	Hypersonic parameter σ
SQT	Intermediate quantity in transverse curvature correction, eq. I(187)
SWFAC	$(S_w + 1) [7 + 3.4 (S_w + 1)]$
SWFAC2	$1 + 4.5 (S_w + 1)^{0.9}$
SWP1	$S_w + 1$
T	Nondimensional temperature, T/T_0
TE	Free stream temperature, T
TEP	Free stream temperature at which transport property calculations were last done
THETA1	Momentum thickness based on eq. I(185), divided by R0
TR	T^*/T_w , ratio of reference temperature to wall temperature
TRCALC	Logical flag, .TRUE. if transport properties were calculated at the current flow point
TREF	Reference temperature ($^{\circ}\text{K}$)
TREFP	Reference temperature at which transport properties were last calculated
TRP(I)	Array equivalenced to common block /TRPROP/, to permit processing these data in a DO loop
TRPSV(I)	Array used to save the transport properties at the free stream temperature
U	Nondimensional velocity, u/u_s (see US in common block /SS/)

UPRIME	Free stream velocity, u (cm/sec)
VISCR	Viscosity at the reference temperature (poise)
VISCW	Viscosity at the wall temperature (poise)
VR	VISCR/VISCW
WD	Weight factor w in calculation of $d\delta^*/dx$; eq. I(419)
WW	$e^{-\Delta x/a}$ in eq. I(217); weighting factor for \bar{n}_p in the calculation of the averaged correlation parameter, \bar{n}
X	Axial coordinate (cm)
XII	Previous value of XI
XL	Shear parameter ℓ , eqs. I(194)
XP	Previous value of x (cm)
XPXP	$X + XP$
XSAVE(I)	Temporary storage for species mole fractions SAVEC(I)
XSNV	$\min(\bar{n}, 0)$; used in place of \bar{n} in calculation of HTR to ensure the use of a nonpositive value
XTEST(J)	Mole fraction of the J th species used in the last previous calculation of transport properties
XX	Axial coordinate variable in numerical integration from x_0
YZ(L)	Radius of the L th profile (cm)
ZERO	0.

4.75 Subroutine BICALL

FINAL Logical subroutine argument; **.FALSE.** when BLAYER is called during an intermediate calculation of the nonequilibrium integration, **.TRUE.** for the final calculation of conditions at the end of each successful integration step

GJ(J) Species concentrations in moles/g

J DØ index over the species

4.76 Subroutine BXSECT

I First index of species pair in V array

J Second index of species pair in V array

K Value of KQ for step L of cross section computations

L Index of step in cross section computations which is currently being considered

M Number of sets of constant factors to be used in computations, for option KQ = 14 only

MADD Index used in locating species pairs in V array

MM MM + 1 and MM + 2 are the locations of the current species pair in the V array

MV Index of last parameter in V array for step L-1 of the computations

4.77 Subroutine COMM

ALGJ(J) $\ln \gamma_j$, natural logarithm of the Jth species concentration

CLKF	$\ln k_f$, natural logarithm of the forward rate constant
CLKR	$\ln k_r$, natural logarithm of the reverse rate constant
CLTBF	$\ln (\sum p \gamma_j)$, where the sum is over the third-body species in a reaction
CLZ2	$\ln (Z2)$
CLZ3	$\ln (Z3)$
EF	ϵ_{fi} in eq. I(32lc)
ER	ϵ_{ri} in eq. I(32lc)
GJ(J)	Concentration (γ_j) of the Jth species (mole/g)
I	DØ index over the reactions
ICHG	Equal to 1 for a neutral species, equal to 2 for a charged (ionic) species
ICØUNT	Counter for number of times CØMM has been called in a run
ICYCLE	Interval in ICØUNT at which ISW5B is set equal to 1 to provide diagnostic dumps in subroutine CØMM, EXACT, RNKT, and PRTA
IPAI	Index of atomic species produced in current reaction for KTF(I) = 4
J	DØ index over the species
KF	Indicator for temperature dependence of forward reaction rate constant; for KF = 1, k_f is a function of T; for KF = 2, it is a function of T_0 . For other values of KF, see discussion of KTF input in Sec. 2.3 of Vol. II.

KR	Indicator for temperature dependence of reverse reaction rate constant; for KR = 1, k_r is a function of T; for KR = 2, it is a function of T_e
LØGC	Logical indicator for method of calculation of PICH(I); for LØGC = .FALSE., PICH(I) is computed as $P_i \chi_i$; for LØGC = .TRUE., it is calculated as $-\exp[\ln P_i + \ln (1 - \chi_i)]$ to avoid possible underflow of the exponential function in $P_i = \exp(\ln P_i)$
ØNE	1.
QELAS	Energy transfer to the electron gas as a result of elastic collisions ($\text{cal}/\text{cm}^3 \text{ sec}$)
QF	\dot{q}_{fi} in eq. I(32lb)
QR	\dot{q}_{ri} in eq. I(32lb)
R	Effective nozzle radius for loss of resonance radiation
RCALC	Logical indicator; if .TRUE., R has already been computed for current flow point
RKF	k_f , forward rate constant
RKR	k_r , reverse rate constant
SUMF	$\sum_{j=1}^n \nu_{ij} (\ln \rho + \ln \gamma_j)$
SUMR	$\sum_{j=1}^n \nu'_{ij} (\ln \rho + \ln \gamma_j)$
TAU	Optical thickness based on R for resonance radiation
TEP	Electron temperature, T_e ($^{\circ}\text{K}$)

TEQ	T_e^3
T6	$\sum_{j=1}^n \nu_{ij} \ln \gamma_j$
T7	$\sum \gamma_j$ over the third-body species for a reaction j
VEJ	Collision frequency factor ν_{ei} in the calculation of energy transfer to the electron gas by elastic collisions
VIC	Electron-ion elastic collision frequency (ν_{ei}) divided by the ionic species concentration γ_j
VIC1	$(8/3) \sqrt{\pi/m_e} e^4 N_0 / (2k)^{3/2}$, where m_e is the electron mass (g), e is the electronic charge (esu), N_0 is Avogadro's constant, and k is Boltzmann's constant
VIC2	$k^3 / (\pi e^6)$
VNC	$(4/3) \sqrt{8k / (\pi m_e)} N_0$
VNCC	Electron-neutral collision frequency (ν_{ej}) divided by the neutral species concentration (γ_j)
Y	Radius of first channel profile, cm
Z	Radius of second channel profile, cm
Z1	$\ln u$, natural logarithm of flow velocity in cm/sec
Z2	$R_0 T'$, where $T' = T$ for $KR = 1$ and $T' = T_e$ for $KR = 2$ ($\text{cm}^3 \text{ atm/mole}$)
Z2F	$R_0 T_0$ ($\text{cm}^3 \text{ atm/mole}$)
Z3	$\rho R_0 T'$ (see Z2)

4.78 Subroutine CXSECT

I(J) New index of species for which old index is J

J Second index of species pair

L First index of species pair

M Temporary variable

4.79 Subroutine DERIVS

CPJE Temporary storage for specific heat of the electrons

CTSAVE Temporary storage for the nondimensional gas temperature T/T_0

DELI1(L) δ_{i1}^* for the Lth profile (eq. I(424))

DELI2(L) δ_{i2}^* for the Lth profile (eq. I(424))

DELØ1(L) δ_{o1}^* for the Lth profile (eq. I(424))

DELØ2(L) δ_{o2}^* for the Lth profile (eq. I(424))

DEN Denominator in eq. I(424)

HJE Temporary storage for nondimensional enthalpy of the electron gas (H_e/R_0T_0)

I DØ index over species

ITRY Counter for iteration to obtain self-consistent solution for the boundary layer displacement thickness (Section 7.6 of Vol. I)

K DØ index over profiles

SENTE Temporary storage for nondimensional entropy of the electron gas (S_e^0/R_0)

4.80 Subroutine DSMSOL

A Matrix initially containing the coefficients and constants for a system of linear equations to be solved:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1i}x_i = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2i}x_i = b_2$$

.

.

.

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ii}x_i = b_i$$

The coefficients a_{jk} are stored in the locations $A(j,k)$, for $j=1$ to i and $k=1$ to i . The constant terms b_j are stored in $A(j,i+1)$, for $j=1$ to i . The solution values x_k are written into the locations $A(k,i+1)$ for $k=1$ to i , that is, the locations originally occupied by the b_j .

I	Number of equations in the system to be solved
J	DØ index running from 1 to I
K	DØ index running from 1 to I
KS	Error indicator; nonzero value indicates singular matrix of coefficients
L	Counter for matrix elements during rewriting of the matrix in close-packed form
LM	First index for close-packed form of matrix
LN	Number of columns already filled in close-packed matrix
LN1	Column index (second index) in close-packed matrix
M1	I + 1

N First dimension of A in the dimension statement in the calling routine

N1 N + 1

RNAME Routine name for calling DUMP routine

4.81 Subroutine DUMP

RNAME Name of routine from which DUMP was called

4.82 Subroutine DUMPER

No local variables.

4.83 Subroutine ELCØND

I Species index

SIGMA Electrical conductivity in mhos/cm

SSIG $(e/k)^2 / \sqrt{T}$

4.84 Subroutine ELTIME

CALLED Logical indicator, initially .FALSE., set to .TRUE. when ELTIME is first called

DET Elapsed time since last printed time (seconds)

ET Elapsed time from beginning of execution (seconds)

ETM Elapsed time from beginning of execution (minutes)

ETØ ET argument value passed from calling routine

ETP Previous value of ETØ

I Elapsed time from start of execution in milliseconds

IP Control for printing the time; zero value suppressed the print

4.85 Subroutine EPART

EF Energy gained by the electron gas in the forward direction of a reaction (cal/mole)

ER Negative of the energy gained by the electron gas in the reverse reaction (cal/mole)

E0 Reaction energy to be partitioned between radiative losses and heating of the electron gas, ϵ_0 (cal/mole)

IR Reaction index in the list of reactions for the current gas model

IT Indicator (ITR) of the rule for partitioning the reaction energy in the current reaction

QF Energy lost by radiation in the forward reaction (cal/mole)

QR Negative of energy lost by radiation in the reverse reaction (cal/mole)

R0 Universal gas constant, cal/mole-°K

TE Electron temperature (°K)

THR $3R_0/2$, where R_0 is the universal gas constant in cal/mole-°K

4.86 Subroutine EOCALC

CTSAVE Variable used to save and restore the initial value of the nondimensional temperature CT

I DØ index

IL I + ISC

J DØ index
 K DØ index
 L I + ISC
 N Iteration counter for Newton-Raphson solution of equations I(233)
 P Pressure (atm)
 RNAME Routine name for calling DUMP routine
 T Temperature (°K)
 ZA Newton-Raphson correction factor $(1 + h_n^x)$ to mole fractions, eq. I(235)

4.87 Subroutine EXACT

AAA(L,M) Array of coefficients and constants for system of linear equations
 DGJ(J) $d \gamma_j / dx$
 ECØEF $1 / (\rho u R_0 T_0)$
 GJ(J) γ_j , species concentrations, moles/g
 I DØ index
 J DØ index over species
 K I - ISC
 NE Number of equations to be solved for rates of change of dependent variables during the nonequilibrium integration
 SUMTDG $\sum_{j=1}^n \frac{T}{T_0} \frac{d \gamma_j}{dx}$ for chemical nonequilibrium models;
 $\frac{T_e}{T_0} \frac{d \gamma_e}{dx} + \sum_{j=2}^n \frac{T}{T_0} \frac{d \gamma_j}{dx}$ for electronic nonequilibrium models

$$SI \quad -1 + \nu_{i-c}^* + \frac{H_i}{R_0 \bar{T}} - \sum_{j=1}^c \bar{\nu}_{i-c,j} \frac{H_j}{R_0 \bar{T}}$$

$$l \quad (u/u_s)^2/w_0$$

4.88 Function EXP

EXP Value of function

X Subroutine argument

4.89 Subroutine FINDX

A Geometric area ratio (from calling routine)

AR Geometric area ratio based on a trial value of X

DADX Derivative of geometric area ratio with respect to X (cm⁻¹)

DZDX(I) Slope of Ith profile in channel

ERA Error in geometric area ratio AR at current value of X

ERRL Smallest negative error in AR for any previous trial value of X

ERRU Smallest positive error in AR for any previous trial value of X

IENTRY Indicator of entry point; 1 for FINDX, 2 for FINDXC

MBL Index (1 or 2) of the profile whose ordinate varies most rapidly downstream of the throat in a channel

N Iteration counter

RNAME	Routine name for call to DUMP routine
UPDOWN	Indicator; -1.0 if upstream solution is desired, 1.0 if downstream solution is desired
UPPER	Logical indicator; .TRUE. if a positive ERA has been found at some trial value of X
V	Trial value of X (cm)
VL	Trial value of X corresponding to error ERRL
VØ	Trial value of X in previous iteration step
VU	Trial value of X corresponding to error ERRU
X	Final solution for X (cm)
Z(I)	Ordinate of the Ith profile in a channel

4.90 Subroutine FROZEQ

AR	Common variable, used here for the geometric area ratio
AS	Sound speed (cm/sec)
BLVAR(I)	Array equivalenced to the contents of common block /BLOUT/
BLVARØ(I)	BLVAR values at preceding flow point
BLVARs(I)	BLVAR values at the current regular flow point; used in interpolating at a model point
CLNMA	Logarithm of the molecular weight in the reservoir
CTSAVE	Nondimensional temperature at the regular flow point following a model point

CTSTAR	The nondimensional temperature (CT) value at the flow point preceding the throat, in solutions including the boundary layer
CT1	Second most recent trial value of CT in the iteration to find the conditions at a model point
CT2	Most recent trial value of CT in iteration to find the conditions at a model point
CXSAVE	X coordinate of the regular flow point following a model point
DELBLØ(I)	For I = 1, 2, the nondimensional boundary layer displacement thickness for the Ith profile at the preceding flow point
DEBLP(I)	For I = 1, 2, the nondimensional boundary layer displacement thickness for the Ith profile at the point before the preceding flow point
DELT	Decrement in nondimensional temperature CT in calculation of frozen throat conditions
DELTSV	Variable used to save the input value of DELT1
DELTV	Current decrement in nondimensional temperature CT
ERRX	Position error in iteration to determine the conditions at a model point
FLAG	Indicator for upstream and downstream regions, for use by the geometry subroutines. Negative value indicates an upstream position, positive value a downstream position

F1,F2,F3	Values of nondimensional mass flux in iteration to determine the frozen throat conditions
GJ(J)	Species concentrations, mole/g
I	DØ index
IPASS	Indicator for iteration to take the boundary layer displacement thickness into account in the determination of reservoir conditions. IPASS = 1 causes a return to the main program after the throat has been reached
IRSTRT	Counter for number of times solution has been restarted with a lower value of W to suppress instability due to coupling of the boundary layer and the inviscid solution
ISØLN	Indicator for type of flow solution: ISØLN = 1 Frozen solution ISØLN = 2 Equilibrium solution
ISØNIC	Indicator for region of flow solution: ISØNIC = -1 Upstream region ISØNIC = 0 Sonic point ISØNIC = 1 Downstream region
ISWTCH	Counter for number of flow points beyond the throat
ITS	Index of the next specified test section diameter (TSDIAM)
J	DØ index over gas species
K	DØ index
LASTPT	Indicator for last point in flow solution (at CX = CXMAX)
M	DØ index over channel profiles
MØDLPT	Logical indicator; if .TRUE., a model point is being calculated
N	DØ index

NMPI	Counter for number of iterations to determine the conditions at a model point
ØNE	1.0
SKIP	Logical variable used to indicate that conditions at a model point have just been determined
SØLN(I,J)	Hollerith array
SWTCH	Logical indicator; if .TRUE., FRØZEQ has begun taking the displacement thickness into account in calculations of the flow-point position
S1	$-\frac{1}{W_0} \sum_{j=1}^n x_{j0} \ln x_{j0}$
S2	$-(\ln p_0)/W_0$
TSTEPC	Logical indicator for special control of temperature decrement far downstream, to prevent distance of successive flow points from becoming too large
T1,T2,T3	Nondimensional temperature values in iteration to determine frozen throat conditions
XFRAC	Distance from preceding flow point to model point, divided by distance from preceding flow point to current point
XIAST	Position of preceding flow point (cm)
XMØDEL	Model-point position (cm)
XMØD1	Position of the next model point in the geometric sequence of model points (cm)
XØ	Position of the preceding flow point (cm)
XP	Position of the point before the preceding flow point (cm)

X1	Position of the flow point corresponding to the second most recent trial value of CT in the iteration to find the conditions at a model point (cm)
X2	Position of the flow point corresponding to the most recent trial value of CT in the iteration to find the conditions at the model point (cm)
Z	$F1 - F2 + F3 - F2$
4.91 <u>Subroutine GEOM</u>	
AN	Effective area ratio corresponding to the current trial value of ρ , based on the area ratio-density relation, eq. I(383)
A2	Square of the effective area ratio
ARG	$2C(A_e - 1) / \alpha$
DADR	$\rho_0 dA_e / d\rho$, derivative of the effective area ratio with respect to nondimensional density
DD	Correction in nondimensional density
DELRHØ	Absolute value of the difference between ρ / ρ_0 and the nondimensional density at the throat, based on approximate solution near throat
DELSTP	Displacement thickness at the throat (cm)
D2LA	$[d^2(\ln A_e) / dx^2]$, the second derivative of the logarithm of the effective area ratio at the throat
J	DØ index over the profiles in a channel
N	Iteration counter
NSUJ	Number of upstream sections in the curvefit to a profile

ØMZ	1. - Z
ØPHA	$1 + \alpha/2$, where α is the exponent in the density-area ratio relation I(383)
RNAME	Routine name for call to DUMP
SGN	-1 if dA_e/dx is positive, +1 if dA_e/dx is negative
Y	Consistency check on x and ρ values; if positive, x and ρ data are inconsistent
Z	$(\rho/\rho_0)^\alpha$

4.92 Subroutine GEØMAR

ARATIØ	Geometric area ratio
DAB	Difference between X and the x-coordinate of the center of a circular profile section
DERIVA	dA_g/dx , derivative of geometric area ratio (cm^{-1})
DRV(I)	Slope (derivative) of ordinate for Ith profile
DYDX	Slope of ordinate for first profile, after rescaling area
DZDX	Slope of ordinate for second profile, after rescaling area
I	Index of two profiles
IENTRY	Index of entry point: 1 - GEØMAR; 2 - GMAR; 3 - GMAR2; 4 - GMAR3
ISJ	Shape index for a profile section

J	Index for the sections in a profile
K	Index for the profile section in which X lies
NSM1	Number of sections in a profile minus 1
Q	Difference in ordinate between a point on a circular profile section and the circle center (cm)
RAD(I)	Ordinate of the Ith profile (cm)
RAT	Ratio of a profile ordinate to the ordinate at the throat
SN(I)	-1.0 for I = 1, + 1.0 for I = 2
SQTRSA	Square root of area rescaling factor RSA
S1(I)	Unrescaled geometric area ratio; or ratio of ordinate to ordinate at the throat for Ith profile
S2(I)	Derivative of geometric area ratio; or ratio of derivative of ordinate to ordinate at the throat for Ith profile (cm ⁻¹)
X	Axial coordinate in nozzle (cm)
Y	Ordinate of first profile (cm)
Z	Ordinate of second profile (cm)

4.93 Subroutine INGAS

CAPQ(K)	Q_k , number of gram-atoms of the kth chemical element per mole of the cold gas
CCI(I)	Intermediate variable in the calculation of the molecular weights of species from atomic weights of elements

I	Species index
IL	Index running from c + 1 to n
J	Element index
K	Index running from 1 to c
L	Index running from c + 1 to n
SUM	Dummy variable used for accumulating summations

4.94 Subroutine INIT

I	Species index in the current gas model
II	Species index in the master list of species
J	Index of additional vibrational modes in triatomic molecules
K	Number of electronic levels for a species
L	Index of the electronic levels for a species
SPRP(M,II)	Array containing species property data in common /SPEC/
T1	Reservoir temperature $T_0(^{\circ}\text{K})$
T2	T_0^2
T3	T_0^3
T4	T_0^4

4.95 Subroutine KANDMU

I	Species index
IL	I-1

(1)

J Species index

L Index of property being calculated; L = 1 for translational thermal conductivity, L = 2 for viscosity

R Weighting factor used in computing $\bar{a}^{(\alpha)}$ from eq. I(86)

U Variable used to accumulate numerator in eq. I(86)

V Temporary variable used in computing denominator in eq. I(86)

W Temporary variable used in computing numerator in eq. I(86)

Y Temporary variable, used for accumulating denominator in eq. I(86) and numerator in eq. I(85)

(1) ZK(L) ZK(1) $\cdot \sqrt{T}$ is the translational thermal conductivity in milliwatts/cm-°K, ZK(2) $\cdot \sqrt{T}$ is the viscosity in millipoise

4.96 Subroutine KINT

I Species index

II I-1

J Summation index for sum over species pairs with J < I

L Index of property being computed (see discussion of KINT in Sec. 3.28)

NI Species index

NJ Summation index for sum over species pairs with NJ > NI

(1)

N1 N + 1

ZKINT ZKINT \sqrt{T} is the internal thermal conductivity in milliwatt/cm-°K

4.97 Subroutine LIST

CAPQ(I) Q_k , number of gram-atoms of the kth chemical element per mole of the cold gas

CØM1(I,J) Hollerith array for output (see DATA statement)

CØM2(I,J) Hollerith array for output (see DATA statement)

CXMAXI Distance beyond throat at which case will be terminated (inches)

I DØ index

J DØ index

K DØ index

KDQ1 KDIM + 1

NWXI Number of specified distances WXI from the leading edge of a wedge model

SBJ(J) b_j , eqs. I(51)

SHJAP(J) Formation enthalpy of the Jth species, cal/mole

THEVP(J) Characteristic vibrational temperature for the Jth species, °K

WØRD(I,J) Hollerith array for output (see DATA statement)

4.98 Subroutine MATINV

A	Original matrix; replaced by inverse
AA	Temporary storage used in interchanging columns
B	Temporary storage used in interchanging rows
BB	Temporary storage used in reduction of off-diagonal elements
BIG	Pivotal element in a column (largest absolute magnitude)
I	DØ index
IV(K)	Index array for recording interchanges of rows
J	Temporary storage used in interchanging IV elements; also DØ index
JA	Index of current row; reset to index of next row
JB	Index of preceding row
JJ	Index of current row
K	First index of pivotal element in a column
K1	$K + 1$
L	DØ index
M	Equal to MM
MM	Size of matrix to be inverted
M1	$M - 1$
NN	First dimension of A in dimension statement of calling routine

RNAME Routine name for calling DUMP routine

 4.99 Subroutine MØDEL

ALPHA Ratio of the current estimates of the pressure p_2 and the density ρ_2 behind the shock

CATM Conversion factor from atm to dyne/cm²

CKS1 Parameter characterizing low density effects on the stagnation-point heat flux, ξK^2 (per foot)

CM1 Mean molecular weight of the gas ahead of the shock, g/mole

CTSAVE Temporary storage for nondimensional free stream temperature CT

DET Computer time used in model calculations, seconds

DISC Discriminant in quadratic solution for the pressure behind the equilibrium normal shock

ELN(I) Exponent of Lewis number in Fay-Riddell heat transfer correlation (equilibrium boundary layer, I = 1, frozen boundary layer I = 2)

EMF Temporary storage for electron mole fraction at stagnation point, to permit output of electron concentration

EPS Density ratio $\epsilon = \rho_1 / \rho_2$ across normal shock

EPSØLD Previous estimate of ϵ

EQUALS Hollerith equality sign used in output

ERRHI Most recent positive error in the nondimensional enthalpy in iterative solutions for conditions behind the normal shock and at the inviscid stagnation point

ERRLO	Most recent negative error in the nondimensional enthalpy
ERZ	Current error in the nondimensional enthalpy
ET	Elapsed time since the beginning of the run, seconds
ETO	Elapsed time since the beginning of the run when MDEL is entered, seconds
GJ(I)	Species concentrations, moles/g
HCF	Enthalpy of formation of the gas mixture at the stagnation point
HCN	Conversion factor (RT_0/W_0) from nondimensional enthalpy to enthalpy in cal/g
HD	Nondimensional kinetic energy per unit mass of the gas ahead of the shock, $(u/u_s)^2/2$
HE	Free-stream static enthalpy (cal/g) ; used in flat-plate calculation
HI	Logical variable; if .TRUE., a positive enthalpy error has already been obtained in the iteration
HR	Recovery enthalpy (cal/g) in flat-plate calculation
HRATIO	Ratio of the dissociation enthalpy to $(h_e - h_w)$
HSTAG	Enthalpy at the inviscid stagnation point, cal/g
HSTAR	Reference enthalpy (cal/g) in flat-plate calculation
HW	Gas enthalpy at the wall temperature of the stagnation point model, cal/g

H2	Nondimensional enthalpy behind the normal shock
I	DØ index
ICØUNT	Iteration counter
ISØLN	Index specifying the two types of normal-shock solution (equilibrium ISØLN = 1, frozen ISØLN = 2)
ISS5P1	Number of lines of output required for the mole fractions at the inviscid stagnation point
J	Index of chemical elements
K	DØ index
L	Model-shape index
LIMSCØ(I)	For I = 1, 2, stored values of LMSC for the equilibrium and frozen normal shock solutions
LIM1	Lower limit of implied DØ loop for output of mole fractions
LIM2	Upper limit of implied DØ loop for output of mole fractions
LMSC	Number of data items from the SCØUT array to be printed for the current normal shock solution
LØ	Logical indicator; if .TRUE., a negative enthalpy error has already been obtained in the iteration
N	DØ index
NEL	Number of chemical elements in a species

PC	$4 \rho_1^2 u_1^2, \text{ atm-g/cm}^3$
PRW	Prandtl number at the model wall temperature
PS	Stagnation pressure, atm
PT	$p_1 + \rho_1 u_1^2, \text{ atm}$
P1	Pressure ahead of the shock, atm
P2	Pressure behind the normal shock, atm
QFRE(L)	Stagnation point heat flux times the square root of the nose radius for an equilibrium boundary layer (hemispherical model L = 1, flat-faced model L = 2), Btu/ft ^{3/2} -sec
QFRF(L)	Stagnation point heat flux times the square root of the nose radius for a frozen boundary layer (hemispherical model L = 1, flat-faced model L = 2), Btu/ft ^{3/2} -sec
QP	$p_1 T_2 / T_1$
QS	Factor used in calculating QFRE and QFRF
QSRI	Stagnation point heat flux times the square root of the nose radius based on the SRI approximation, Btu/ft ^{3/2} -sec
QUANT(I)	Hollerith names of output quantities in SCOUT array
RHØSTR	Density at reference temperature in flat-plate calculation
R1	Density ahead of the shock, g/cm ³
R1U12	$\rho_1 u_1^2, \text{ atm}$
R2	Density behind the shock, g/cm ³
SCOUT(I)	Array used for output of model conditions

SHV	Array equivalenced to common block /STAG/ to permit setting these data in the DØ loop over ISØLN
SØD(L)	Shock standoff distance divided by model radius (hemispherical L = 1, flat-faced L = 2)
SQTE	Square root of the shock density ratio, $\sqrt{\epsilon}$
SQ2F(K)	Ratio of the stagnation-point heat flux for the current model geometry to that for an axisymmetric model (two-dimensional model K = 1, axisymmetric K = 2)
THI	T_2 value corresponding to enthalpy error ERRHI, °K
TITLE2	Hollerith data used in output
TLØ	T_2 value corresponding to enthalpy error ERRLO, °K
TSTAR	Reference temperature in flat-plate calculation, °K
T1	Temperature ahead of the shock, °K
T2	Temperature behind the shock, °K
U1	Velocity ahead of the shock, cm/sec
VGP(L)	Velocity gradient parameter (R_p/u_1) (du_e/dx) (hemispherical model L = 1, flat-faced L = 2)

4.100 Subroutine NEWRAP

GJ(I)	Species concentrations, mole/g
I	DØ index; also J + ISC
IL	I + ISC

J	Dø index
K	Dø index
L	I + ISC
LL	ISSNR + 1
M2	ISC + 2
NØDEPS	Logical indicator; .TRUE. when the gas model contains no dependent species
RNAME	Routine name for calling DUMP routine
ZB	Correction factors $1+h_n^r$ to the mole fractions, eq. I(260)
ZC	Correction factor $1+h_{c+1}^r$ to the pressure

4.101 Subroutine NEXTMP

ITS	Index of the next specified test section diameter
ØNE	1.0
XMØD1	Coordinate of the next model point in the geometric sequence of model points, cm
XMØD2	Coordinate of the model point at the next specified test section diameter, cm
XMØDEL	Coordinate of the next model point, cm

4.102 Subroutine NØNEQ

AB	Temporary storage location used in interchanging rows of β_{ij} matrix during determination of matrix rank ^{ij}
----	--

ADCH Absolute value of a $\delta\chi_i$ value

ALOGT $\ln(10000)$

BLNEAR(M) Array equivalenced to the first 10 locations of common block /BLNE/

CA BETA(M,I)/BETA(IX,I)

CAISV(I) Storage for the correct values of CAI(I) to allow resetting CAI(I) when the artificial increase of reaction rate constants has been used to prevent premature switching from the perturbation technique to the numerical integration

CI(K) Array specifying elemental composition of the gas mixture in the reservoir; CI(K) is the number of gram-atoms of the Kth element per gram of mixture

CJ Number of gram-atoms of an element per gram of mixture at the end of a successful integration step

DCA(K) $\frac{1}{(\gamma_k')^2} \sum_{j=1}^C \delta c_j A_{jk}$ in eq. I(346c)

DCHMAX Largest $|\delta\chi_i|$ for any reaction

DCHMIN Smallest $|\delta\chi_i|$ for any reaction

DCI(K) δc_k , correction in concentration of Kth chemical element required at the end of an integration step (eq. I(344), g-atom/g)

DELB(L) Nondimensional boundary layer displacement thickness for the Lth profile, δ_L^*/R_0 , at the start of an integration step

DELT	Nondimensional temperature interval in the perturbation solution, $ \Delta T /T_0$
DELTSV	Input value for DELT1
DQEM	Maximum allowable change in the energy transfer to the electron gas in one integration step
DXM	Largest allowable step size for the first integration step after the switch from the perturbation solution
DXOLD	Step size Δx in the preceding integration step (cm)
DXSAVE	Temporary storage for saving and restoring the integration step size when a reduced step is used to calculate the conditions at a model point (cm)
GJ(J)	Concentration of the Jth species, γ_j , mole/g
GSQ(J)	Square of GJ(J)
HDXSAV	One-half of DXSAVE, cm
I	DØ index
ICØUNT	Counter for number of times a step has been restarted with reduced step size
IERR	Error exit indicator (see subroutine listing)
IF(N)	Array for storing and printing out IFAIL values for successive restarts of a step (see glossary for common block /TNERK/)
II	I + ISC
IMAX	Index of the reaction with the largest $ \delta \chi_i $

IRA	Counter for number of times the artificial increase in rate constants has been used to prevent premature switching from the perturbation technique to numerical integration
ISMCP1	$n - c + 1$
ISTEPS	Counter for number of integration steps
ISUBD	Counter for number of times the temperature interval ΔT has been cut in half in an attempt to find a point in the perturbation solution satisfying the second inequality of eq. I(381)
ISUPR(I)	Array of indices of the reactions which are suppressed when the decrease of a minor species is found to control the integration step
IS5	Input value of ISW5B
ITAPE	Number of records written on the binary output tape since subroutine THRØAT was called
ITS	Index of the next model point defined by a specified test section diameter
IX	Rank of the β_{ij} matrix
I2	Index used in calculating the rank of β_{ij}
J	Index used for various purposes
K	Index used for various purposes
IA	Index used in calculating the rank of β_{ij}
L	DØ index
M	Index used in calculating the rank of β_{ij}

N	Index used in calculating the rank of β_{ij}
NSUPR	Number of reactions suppressed when the decrease of a minor species is found to control the integration step
ØNE	1.0
QEØLD	Energy transfer to the electron gas at the preceding flow point, cal/cm ³ sec
RATIØ	Ratio of the smallest and largest $ \delta\chi_i $ values
RATIØ2	Factor by which the forward rate constant is increased in the artificial increase of a reaction rate to prevent premature switching from the perturbation technique to the numerical integration
RNAME	Subroutine name for call to DUMP routine
SCD	Factor by which the step size Δx is divided when an integration step fails and has to be restarted
SDGJ(J)	Change in the concentration γ_j of the Jth species in an integration step
TENTH	10000.
TEP	Electron temperature (°K)
TP	Gas temperature (°K)
TSZ	Temporary storage for TPRINT
XMØDEL	Position of next model point, cm
XMØD1	Coordinate of the next model point in the geometric sequence of model points, cm

XMSET Logical indicator, set to .TRUE. when NEXTMP is first called to determine XMDEL

ZZZ Temporary storage

4.103 Subroutine NRMAX

DELT Decrement of nondimensional temperature

F1,F2,F3 Mass flux values in iteration to determine equilibrium sonic flow conditions

GJ(J) Species concentrations, mole/g

I DØ index over species

J DØ index over species

T1,T2,T3 Nondimensional temperature values in iteration to determine the equilibrium sonic flow conditions

Z $F1 - F2 + F3 - F2$

4.104 Subroutine ØUT1

ANAM Hollerith word "ARAT"

ARAT Geometric area ratio

ASØLN(I,J) Hollerith array containing names for types of solution (see DATA statement)

ASTRSK Hollerith word filled with asterisks

CHDIMS(L) Transverse dimensions of a channel at the current flow point, inches

CP Molar heat capacity at constant pressure, divided by R_0

DARAT	Derivative of the geometric area ratio
DEL	Electron density, electrons/cm ³
DELSTP(L)	Displacement thickness on the Lth profile, cm
DGJ(I)	$d\gamma_i/dx$, derivatives of species concentrations, mole/g-cm
DLNDX(I)	$d \ln \gamma_i / dx$ for the Ith species
DNAM	Hollerith work "DIAM"
EMF	Mole fraction of the electrons
FAREA	Effective cross-sectional area of the flow at the throat, cm ²
FLX	Mass flux, lbm/ft ² -sec
FVØ2(I,J)	Array used for output of SN and XSN
GAMMA	Ratio of specific heats, $\gamma = c_p/c_v$
GJ(I)	Species concentrations, mole/g
HØUT	Enthalpy, Btu/lb
HW	Enthalpy at the nozzle wall, Btu/lb
HO	Stagnation enthalpy, Btu/lb
I	DØ index
IBL	Index such that FVØUT(IBL) is the boundary layer displacement thickness on the nozzle wall or on the broad face of a channel
IFLAG	Indicator for entry points to subroutine IFLAG = 1 Entry ØUT1 IFLAG = 2 Entry ØUT2(ISØIN)

II	Index 7 + 10*I
INEQP1	INEQ + 1
ISØLN	Indicator for type of solution ISØLN = 1 Frozen ISØLN = 2 Equilibrium ISØLN = 3 Nonequilibrium
ISS5P1	(ISS-1)/5+1, number of lines of output re- quired for printing the species mole frac- tions
IS6	ISW6B
IX	Counter of records written on tape 8 for a given flow solution
I1	Index of the first FVØUT(I) element to be printed on a line
I2	Index of the last FVØUT(I) element to be printed on a line
I6	Counter; number of flow points printed since last output of reaction rate data
J	DØ index
K	DØ index
L	DØ index
LIM1	Index of first species mole fraction to be printed on a line
LIM2	Index of last species mole fraction to be printed on a line
PØUT	Pressure, atm
PZ	3.1415927

RØUT	Density, lb/ft ³
SØUT	Entropy, Btu/lbm-°R
STEP(I)	Hollerith array (see DATA statement)
STFAC	$1 / [\rho u(h_0 - h_w)]$, (Btu/ft ² -sec) ⁻¹
TFLØW	Total mass flow, lb/sec
TNNAM(I)	Hollerith array of variable names (see DATA statement)
TNØUT(I)	Array used for output of quantities for electronic nonequilibrium model
TØUT	Temperature, °K
TYPSELN(I,J)	Hollerith array used to indicate whether non-equilibrium solution is based on perturbation technique (INEQ = 0) or numerical integration (INEQ = 1)
UØUT	Velocity, ft/sec
VARNAM(I)	Array of Hollerith names for output variables (see DATA statement)
VARN2(I)	Array of Hollerith names for N, XSN
WNAM	Hollerith word "WIDTH"
Y	Distance from nozzle axis to nozzle surface (or to first profile in the case of a channel)
Z	Distance from axis to second profile of a channel

4.105 Subroutine PERT

I	DØ index
J	DØ index

K DØ index

Z Temporary storage used in computing sums

Z1 Temporary storage used in computing sums

4.106 Function PIØMEG

I Index of entries in cross section table

PIØMEG Cross section value returned to the calling routine

TE Electron temperature, °K

4.107 Subroutine PRØP

J DØ index over species

RNAME Routine name for call to DUMP

S4 $\frac{1}{R_0} \sum_{j=1}^n \gamma_{j0} s_j^0$

4.108 Subroutine PRTA

AS Speed of sound, cm/sec

4.109 Subroutine PUTQIN

I First index of species pair

IDBG Count to determine when debug output is produced

J Second index of species pair

K Option KQ to be used in step L of the computations

KK Type of cross section; first index of Q(KK,I,J)

L	Index of current step in cross section computations
IQ	Index of species pair in IQ,JQ array
IQ1	Index of first species pair in IQ, JQ array to be used in step L of computations
IQ2	Index of last species pair in IQ, JQ array to be used in step L of computations
M	Index of first parameter in V array to be used in step L of computations
$\phi M(KK)$	Variable for temporary storage of cross sections computed in current step. $\phi M(1) = \bar{\Omega}(1,1)$, $\phi M(2) = \bar{\Omega}(2,2)$, $\phi M(3) = B * \bar{\Omega}(1,1)$
WRØTE	WRØTE = .TRUE. indicates that debug output has been produced in the current call to PUTQIN
X	Electron partial pressure in atmospheres
	4.110 <u>Subroutine QCØUL</u>
C	Control variable. For $C > 0$, $\ln(f \wedge)$ is computed from eqs. I(100); for $C \leq 0$, $\ln(f \wedge)$ is set to 1
$\phi M1$	$0.8Q_C$ (see eq. I(100b))
X	Electron partial pressure in atmospheres
V	$20.91Y = (f \wedge)^2$ (see eqs. I(100c) and I(100d))
	4.111 <u>Subroutine QEX</u>
A	Coefficient A in eq. I(104) for the exchange cross section

B Coefficient B in eq. I(104) for the exchange cross section
 $\sigma_M(K)$ Contribution to cross sections from option;
 $\sigma_M(1) = \bar{\Omega}(1,1)$ $\sigma_M(2) = \bar{\Omega}(2,2)$
 $\sigma_M(3) = B * \bar{\Omega}(1,1)$
 V(M) Input parameters VV for option
 Y Quantity $A - \frac{1}{2} B \ln(T/M)$
 ZM Molecular weight

4.112 Subroutine QEXP

A Coefficient A/k in $^{\circ}\text{K}$ for the exponential potential
 ALPHA Quantity α
 N1 Position of first entry in tabulated collision integrals for the exponential potential
 $\sigma_M(K)$ Computed cross section values (see subroutine QEX)
 RHO Characteristic length ρ for the exponential potential in \AA
 V(M) Input parameters VV for option

4.113 Subroutine QINTRP

A Value of independent variable to be used
in interpolation

B(J) Interpolated value of Jth dependent var-
iable

I Interpolation is performed between entries
I and I + 1 of the table

J Index of dependent variable

N Number of entries in table

N1 Location of first entry in table

N2 N2 + 1 is the location of the last entry
in table

4.114 Subroutine QLJ

EPSLN Well depth parameter ϵ/k for the Lennard-
Jones potential, in $^{\circ}\text{K}$

N1 Position of first entry in tabulated
cross sections for the Lennard-Jones
potential

$\phi\text{M}(K)$ Computed cross section values (see
subroutine QEX)

SIGMA Collision diameter σ for the Lennard-Jones potential, in Å

TSTAR Nondimensional temperature, kT/ϵ

V(M) Input parameters VV for option

4.115 Subroutine QMIX

I First index of species pair

J Second index of species pair

K Type of cross section, i.e., first index of $Q(K,I,J)$

IQ Index of species pair in IQ, JQ array

IQ1 Index in IQ, JQ array of first species pair to be used in computation

IQ2 Index in IQ, JQ array of last species pair to be used in computation

SQT(K,I) $\frac{1}{2} \sqrt{Q(K,I,I)}$

4.116 Subroutine QREPP

ITL Location in ϕ MEGAL array where data are stored

ϕ M(K) Computed cross section values (see subroutine QEX)

VV(M) Input parameters VV for option

4.117 Subroutine QSAME

C Constant by which previously computed cross sections are multiplied

I First index of species pair for previously computed cross sections

J Second index of species pair for previously
computed cross sections

$\phi M(K)$ Computed cross section values (see subroutine
QEX)

VV(M) Input parameters VV for option

4.118 Subroutine QTAB

A Constant multiplying factor for tabulated
cross sections

N1 Location of first entry in tabulated cross
section data

NL Number of entries in tabulated cross section
data

$\phi M(K)$ Computed cross section values (see subroutine
QEX)

V(M) Input parameters VV for option

4.119 Subroutine Q11

LQ1 Location in IQ, JQ list of the first species
pair to which the ramp function is to be ap-
plied

LQ2 Location in IQ, JQ list of the last species
pair to which the ramp function is to be applied

T0 Temperature in $^{\circ}K$ beyond which cross sections
are set to zero

T1 Temperature in $^{\circ}K$ beyond which cross sections
are unchanged

VI(M) Input parameters VV for option

VV(M) Input parameters for subroutine Q14

4.120 Subroutine Q12

I First index of first previously computed species pair

J Second index of first previously computed species pair

K First index of second previously computed species pair

L Second index of second previously computed species pair

M Index indicating cross section type

ϕ M(M) Computed cross section values (see subroutine QEX)

VV(N) Input parameters VV for option

4.121 Subroutine Q13

C Constant by which cross section is multiplied

I First index of species pair

J Second index of species pair

K1 Type of cross section to be computed

K2 Type of cross section to be used in computation

IQ Location of species pair in IQ, JQ list

IQ1 Location in IQ, JQ list of first species pair to be used

IQ2 Location in IQ, JQ list of last species pair to be used

VV(M) Input parameters VV for option

4.122 Subroutine Q14

C2 Constant applied to $\bar{\Omega}$ (2,2)

C3 Constant applied to $B^* \bar{\Omega}$ (1,1)

I First index of species pair

J Second index of species pair

IQ Location of species pair in IQ, JQ list

IQ1 Location in IQ, JQ list of the first species pair to be used with the set of multiplying factors

IQ2 Location in IQ, JQ list of last species pair to be used with the set of multiplying factors

VV(M) Input parameters VV for option

4.123 Subroutine RADIUS

AG Geometric area ratio

AGJ Factor proportional to r^{2j} in eq. I(171)

ITYPE Indicator for type of nozzle geometry
ITYPE = 1 Two-dimensional nozzle
ITYPE = 2 Axisymmetric nozzle
ITYPE = 3 Channel

L Index of profiles (1 for a nozzle, 1 or 2 for a channel)

M	NPR(L)
NPR(L)	2 for L = 1; 1 for L = 2
R	Distance from axis to profile
RATIØ	R/R0
X	Axial coordinate in nozzle or channel, cm
YZ(L)	Distance from axis to Lth profile in a channel
4.124 <u>Subroutine READ</u>	
AL	Number of atoms of the Kth element per molecule of the Ith species, as obtained from the SPRP array
ASYM(IZ)	Hollerith chemical symbol for the element with atomic number IZ
CAPQ(I)	Number of gram atoms of the Ith chemical element per molecule of the cold gas
CP(I,J)	Array for addressing the channel data in common block /CHAN/
CSMW	Molecular weight of one of the cold species
DAL	Error in rounding A to an integer LP
DUM(I)	Array for zeroing common block /TNCE/
EINAME(K)	Hollerith name for the Kth element in one of the cold species
EPRP(I,J)	Array for addressing the element data in common block /ELEM/

EXPMP	Exponent used in computing the factor (FACMP) used to generate a geometric sequence of distances x at which model calculations will be done
FNMP	Number of model points in geometric sequence, minus 1
GPRP(I,J)	Array for addressing the gas model data in common block /MIXT/
HWINØV	Estimate of stagnation enthalpy based on Wino- vich equation, Btu/lb
I	DØ index
IES	Index of an element in a species, in the mas- ter list of elements
IGMV	Number of electronic levels in a species
IGS	Input value of IGAS
II	Index used in place of subscripted integer variables
IPRFL	Index used in place of subscripted integer variable, NPRØFL(J)
ISH	A value of ISHAPE(K,J)
ISKUR	Index of a third-body species in the master list of species
IUPDI	Initial value of IUPD
IZ	Atomic number of an element defined in the input using EEPRP
J	DØ index
JDP1	JDIM + 1

JJ	Index used in place of a subscripted integer variable, JCS(I)
JPS	Index of a species on the product side of a reaction, in the master list of species
JRS	Index of a species on the reactant side of a reaction, in the master list of species
K	DØ index
KK	Index of an element in a cold species, in the master list of elements
L	DØ index
LL	Index used in obtaining the PARAM array from the data stored in the ZPRP array
LP	Integer obtained by rounding AL
NATØM	Number of atoms of an element in a molecule of a species
NELS	Number of elements in a species
NK	Number of third body species for a reaction
NØZZLE	Index of the nozzle profile in the precoded geometry fits
NPSS	Number of product species for a reaction
NRSS	Number of reactant species for a reaction
NSECTD(J)	Number of downstream sections in the profile curvefit for the Jth profile of a channel or nozzle
NSECTJ	Index used in place of dimensioned integer variable, NSECT(J)

NSM1	NSECT(J) - 1
ØNE	1.0
PLUS	Hollerith + sign
RPRP(I,J)	Array for addressing reaction data in common block /REAC/
SBJ(J)	b_j ; eqs. I(51)
SETIGS	Logical indicator; .TRUE. if automatic air model selection option is in effect
SHAPD	Array of Hollerith data for describing the shapes of nozzle profile sections
SHJAP(I)	Formation enthalpy of the Jth species, cal/mole
SPNAME(I)	Array of nominal species names for species defined in the input
SPRP(I,J)	Array for addressing species data in common block /SPEC/
SY1(K)	Array of Hollerith left parentheses
SY2(K)	Array of Hollerith right parentheses
THEVP(I)	Vibrational temperature for the Ith species, °K
TNEP(I,J)	Array for addressing thermal nonequilibrium data in common block /TNE/
WØRD(I,J)	Hollerith data for output of the words "two-dimensional" and "axisymmetric"
WØRD2(I)	Hollerith data for output of the words "gap" and "diam"

ZPRP(I,J) Array for addressing the geometry data in common block /NØZZ/

4.125 Subroutine RESET

I Elapsed time since the beginning of the run in milliseconds

ICPU CPU time remaining before TIME.GØ, milliseconds

IN Entry point indicator: IN = 1, RESET; IN = 2, TIME

IZERØ CPU time remaining before TIME.GØ at call to RESET, milliseconds

4.126 Subroutine RESTMP

AM(I,J) Matrix of coefficients for system of equations in quadratic interpolation for the reservoir temperature

BM(I) Constant terms for system of equations in quadratic interpolation for the reservoir temperature

CØNST $R_0(m_*/p_0)^2$, in which R_0 is the universal gas constant, m_* the known sonic mass flux, and p_0 the reservoir pressure, all expressed in cgs units

CTAPP Previous estimate of reservoir temperature, °K

CTSAVE(I) For I = 1 to 3, successive estimates of the reservoir temperature, the value for I = 1 being the most recent, °K

CTSV Temporary storage for the reservoir temperature estimate used in the current iteration, °K

DCT The difference between the two most recent estimates of the reservoir temperature, °K

DCTAP	Absolute value of the difference between the current and improved estimates of the reservoir temperature, °K
DCTAPØ	DCTAP in the previous iteration step, °K
DMC	Intermediate quantity in quadratic interpolation for the reservoir temperature, °K
D1,D2	Intermediate quantities in linear interpolation for the reservoir temperature, °K
F(I)	For I = 1 to 3, successive estimates of the quantity $W_0 \cdot (SM)^2$, the value for I = 1 being the most recent; W_0 is the molecular weight in the reservoir and SM the nondimensional sonic mass flux
H	Calculated value of reservoir enthalpy, cal/g
HERR	Allowable error in calculated stagnation enthalpy cal/g
HØ	Calculated stagnation enthalpy in previous iteration step, cal/g
I	DØ index
J	Index for shifting data in CTSAVE and F arrays
KS	Indicator for singularity of AM matrix
N	Iteration counter for main iteration
N2	Iteration counter for temperature calculation in enthalpy option
N3	Iteration counter for pressure calculation in enthalpy option

PCGS	Reservoir pressure, dyne/cm ²
PERR	Allowable relative error in reservoir pressure
PIAST	Previous estimate of reservoir pressure, atm
RGAS	Universal gas constant in cgs units
RNAME	Hollerith name for calling DUMP routine
SMERR	Allowable error in sonic mass flux, g/cm ² sec
SMP	Calculated sonic mass flux, g/cm ² sec
TERR	Allowable relative error in reservoir temperature
TLAST	Previous estimate of reservoir temperature, °K

4.127 Subroutine RNKT

DGJ(J)	$d\gamma_j/dx$, derivatives of species concentrations, mole/g-cm
DQMRK	Twice DQMAX; criterion value for allowable changes in energy transfer to the electrons, QDPE, cal/cm ³ sec
FK(J,K)	Array equivalenced to F1(J), F2(J), F3(J), F4(J)
F1(J)	f_1 , derivative of the Jth dependent variable at the start of the integration step
F2(J)	f_2 , derivative of the Jth dependent variable at x_2, y_2 (eq. I(394a))
F3(J)	f_3 , derivative of the Jth dependent variable at x_3, y_3 (eq. I(394b))

F4(J) f_4 , derivative of the Jth dependent variable at x_4, y_4 where y_4 is given by I(394c) or I(405)

GJK(J,K) Array equivalenced to GJ1(J), GJ2(J), GJ3(J), GJ4(J)

GJ1(J) y_1 , the Jth dependent variable at the start of the integration step

GJ2(J) y_2 , the Jth dependent variable at x_2 (eq. I(394a))

GJ3(J) y_3 , the Jth dependent variable at x_3 (eq. I(394b))

GJ4(J) y_4 , the Jth dependent variable at x_4 (eq. I(394c) or I(405))

IFØR Indicator for type of formula used to compute GJ4(J); IFØR=1 for I(394c), IFØR=2 for eq. I(405)

ISTMNT Indicator for locations in subroutine at which diagnostic dumps are written (see listing)

J Index of dependent variables:

$J = 1$ to n	γ_j
$J = n + 1$	T_j
$J = n + 2$	T_e
$J = n + 3$	h_0

K DØ index

LIM Number of dependent integration variables

P(J) Parameter ($-P \Delta x$) for the Jth dependent variable, eq. I(402)

QDPEB Electron energy transfer (QDPE) at the start of the integration step, cal/cm³ sec

SDGJ(J) Change in γ_j over an integration step, mole/g

SDQ(J)	Change in Jth dependent variable over the integration step
S1	Temporary storage in computation of I(398a) for small $P \Delta x$
S2	Temporary storage in computation of I(398b) for small $P \Delta x$
S3	Temporary storage in computation of I(398c) for small $P \Delta x$
TE	Logical indicator, .TRUE. for chemical non-equilibrium models, .FALSE. for electronic nonequilibrium models
T1	Temporary storage in computation of I(398a) for small $P \Delta x$
T2	Temporary storage in computation of I(398b) for small $P \Delta x$
T3	Temporary storage in computation of I(398c) for small $P \Delta x$
XX	$\log_{10}(f_3 - f_2)$
X1	F_1 (eq. I(398a))
X2	F_2 (eq. I(398b))
X3	F_3 (eq. I(398c))
X4	$f_2 - f_1$
Y(K)	Array equivalenced to Y1, Y2, Y3, Y4
YY	$\log_{10}(f_2 - f_1)$
Y1	$f_1 + P_{Y1}$ in I(401)

Y2 $f_2 + Py_2$ in I(401)

Y3 $f_3 + Py_3$ in I(401)

Y4 $f_4 + Py_4$ in I(401)

4.128 Subroutine SHOCK

ALP The quantity $\sin^2 \sigma - B$, where σ is the shock angle

B The quantity $-\left[\frac{M^2+2}{M^2} + \gamma \sin^2 \alpha\right]$, where M is the free-stream Mach number, γ the specific heat ratio, and α the body angle

BETA The quantity $ALP \cdot \sin^2 \sigma + C$

BMF Free-stream Mach number, M

BM2 M^2 , square of the Mach number

BM4 M^4

C The quantity $\frac{2M^2+1}{M^4} + \left[\frac{1}{4}(\gamma+1)^2 + \frac{\gamma-1}{M^2}\right] \sin^2 \alpha$

CD Cosine of the body angle, $\cos \alpha$

C1 $\gamma+1$

C2 $\gamma-1$

C3 2B

D $-\cos^2 \alpha / M^4$

DEL The body angle α ; inclination of the wedge surface to the direction of free-stream motion (degrees)

DELR α in radians

DIS $\sqrt{(ALP)^2 - 4 \cdot BETA}$

FL(I) Array used for communicating results to the calling routine

FL(1) shock angle

FL(2) Mach number behind shock

FL(3) Density ratio, ρ_2/ρ_1

FL(4) Static pressure ratio, p_2/p_1

FL(5) Static temperature ratio, T_2/T_1

FL(6) Total pressure ratio, P_{t2}/P_{t1}

FL(7) Ratio, T_{stag}/T_1

FPU Derivative dF/du of the function $F(u)$

FU $F(u) = u^3 + Bu^2 + Cu + D$

GAM Specific heat ratio γ

IERR Argument whose value is set to 2 when Newton-Raphson iteration to find the shock angle does not converge

KIT Iteration counter

RT $\sin^2 \sigma$, where σ is the shock angle

SD $\sin \alpha$, where α is the body angle

UG Previous trial value of u in solution of cubic equation $F(u) = 0$

UN Current trial value of u in solution of cubic equation $F(u) = 0$

U1 First solution for $\sin^2 \sigma$

U2 Second solution for $\sin^2 \sigma$

V1 $M^2 \sin^2 \sigma$

4.129 Subroutine SIMQ

A(I)	Matrix of coefficients for the system of equations to be solved, stored columnwise with no gaps
B(I)	Constant terms for the system of equations to be solved; these data are replaced by the solution values
BIGA	Pivotal (largest) element in a column
I	Row index
IA	Index of matrix A in back solution
IB	Index of recomputed array B in back solution, containing solution values
IC	Index of array B in back solution
IJ	Index in search for pivotal element in column
IMAX	Index of pivotal element in column
IQS	Index in elimination of variables
IT	Index used for several purposes
IX	DØ index in elimination of variables
IXJ	Index in elimination of variables
IXJX	Index in elimination of variables
IL,I2	Indices in interchange of rows
J	Column index
JJ	Index of the first element in a column of the reduced matrix

JJX	Index in elimination of variables
JX	DØ index in elimination of variables
JY	J + 1
K	DØ index in interchange of rows
KS	Error indicator; set to 1 if matrix of coefficients is singular
N	Number of equations to be solved
NY	N - 1
SAVE	Temporary storage location used in interchanging rows and eliminating variables
TØL	Criterion for value of pivotal element in test for singular matrix

4.130 Subroutine STUNTS

ASTRSK	Hollerith word filled with asterisks
DELTT(I)	Array of preset temperature intervals for table of species thermal properties, °K
H00	Heat of formation of a species in kcal/mole
I	DØ index; also value of IQ(IQ); also value of ISEQ(L)
ICARD	Counter of punched cards containing cross section data
II	Index of preset temperature intervals for thermal property table
IIQJ	Index of the first species of a pair of species whose cross section is computed in the Ith step of the transport property calculation

IT	Counter for lines in the thermal property table with the same value of the temperature increment
IZ	Number of lines in thermal property table
J	DØ index; also JQ(LQ)
JJQJ	Index of the second species of a pair of species whose cross section is computed in the Lth step of the transport property calculation
K	DØ index
KK	KQ(L)
L	Index of steps in the cross section calculation
LIM	1 for species with no thermo fit; 2 for species with a thermo fit
LIM1	1 for species with physical model data; 2 for species without such data
LQ	DØ index
LQ1	Index of the first species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied
LQ2	Index of the last species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied
M	DØ index
MV1	Index of the first parameter (in the V array) used in the Lth step of the edited cross section computation
MV2	Index of the last parameter (in the V array) used in the Lth step of the edited cross section computation

NDT(I)	Number of lines in the thermal property table for which the Ith preset temperature increment is used								
NNQI	Number of species pairs to which the cross section calculated in the Ith step of the calculation is applied								
TPRØP(I,K,M)	Computed thermal properties for the Ith line of the table; M = 1 contains the results of the physical model, M = 2 the thermo-fit results (if any) <table border="0" style="margin-left: 40px;"> <tr> <td>K = 1</td> <td>Chemical potential</td> </tr> <tr> <td>K = 2</td> <td>Enthalpy</td> </tr> <tr> <td>K = 3</td> <td>Specific heat</td> </tr> <tr> <td>K = 4</td> <td>Entropy</td> </tr> </table>	K = 1	Chemical potential	K = 2	Enthalpy	K = 3	Specific heat	K = 4	Entropy
K = 1	Chemical potential								
K = 2	Enthalpy								
K = 3	Specific heat								
K = 4	Entropy								
TT(I)	Temperature value for the Ith line of the thermal property table, °K								
TV	Temperature value, °K								
4.131 <u>Subroutine THERM</u>									
CCPTF	Nondimensional specific heat based on the thermo fit								
CLGT	Natural logarithm of the nondimensional temperature, $\ln(T/T_0)$								
E1	Logical variable, .TRUE. if the routine was entered through the entry point THERM1, otherwise .FALSE.								
J	Species index								
K	Index of vibrational states in a linear triatomic species								
M	Index of the electronic states in a species								

MIX	Logical variable, set to .TRUE. if the species thermal properties are to be calculated by "mixing" results from the physical model and the thermo fit
N	Number of electronic states in a species
SHTF	Nondimensional enthalpy based on the thermo fit
STF	Nondimensional entropy based on the thermo fit
SZ1	Temporary storage for a value required in the entropy calculation for a linear triatomic species
S1	See eq. I(56b)
S2	See eq. I(56c)
S3	See eq. I(56d)
WTHØ	Weight of the physical model results in the "mixed" calculation of species properties
WTF	Weight of the thermo fit results in the "mixed" calculation of species properties
WW	$(S2)^2$
XMTF	Chemical potential based on the thermo fit results
XX	Dummy variable used in calculations of electronic state contributions to species properties
ZT	Difference between the temperature and the switchover temperature CTMXXI from the physical model to the thermo fit, °K
Z1-Z9	Storage for intermediate values in calculations of the vibrational contributions to species properties

4.132 Subroutine THRØAT

AG	Geometric area ratio
I	DØ index
ØNE	1.0
XX	Dummy argument for call to AGSØLN

4.133 Subroutine TRANSP

C(I,J)	Molecular weight dependent factor used in transport calculations (see eqs. 171)
CMSAVE	Mean molecular weight W for the mixture
CPTØT	Nondimensional frozen specific heat at constant pressure for the mixture, $W c_{pf}/R_0$
CSAVE(I)	Array for saving previously computed values of the nondimensional specific heats CCPJ(I)
CTSAVE	Variable for saving previous value of the temperature ratio CT
I	Species index
J	Species index
J1	$J - 1$
P	Gas pressure in atmospheres
S	\sqrt{TTAB}
SSIG	Factor used in computation of electrical conductivity, $(e/k)^2/\sqrt{T}$
TTAB	Temperature in $^{\circ}\text{K}$ for which transport calculations are to be made

ZK(L)	See eq. (173)
ZKINT	See eq. (173)
ZM1(I)	Factor proportional to the molecular weight of species I (see eq. (171))
4.134 <u>Subroutine WEDGE</u>	
A	$(\gamma + 1)/2$, where γ = specific heat ratio
AE	Free-molecule accommodation coefficient for heat transfer
ALMIN	Minimum angle of attack for calculation of free-molecule heat flux (0.01 radian)
ALPHA	Angle of attack in radians
AM	Free-stream Mach number M
AMD	Double-precision Mach number for call to SHOCK routine
AM2	M^2
AND	Input angle of attack in degrees or ANMIN, whichever is larger
ANMIN	Minimum angle of attack for oblique shock calculation (10^{-4} degree)
APR(IA)	Static pressure ratio, p_2/p_1 , based on the oblique shock calculation for the IAth angle
AST(J)	Hollerith array for marking heat fluxes which are at the free-molecule limit with an asterisk
ASTQ(IT)	Hollerith blank or asterisk for ITth column of output table
A2	A^2

A4	A^4
A5	A^5
A7	A^7
C	$[\mu(T^*)/\mu(T_1)] \cdot [T_1/T^*]$; see Sec. 8.2.2, Vol. I
CAPGAM	Angle-of-attack parameter Γ (eq. I(481c) or I(501))
CH	Heat transfer coefficients, C_H
CHD	Coefficient $0.332 A \epsilon \tau N^2 t$ for calculating C_H from S^*
CHDFAC	Coefficient $0.332 A \epsilon$
CPGM	Single-precision value equal to Γ
DELSTW(IT)	Boundary layer displacement thickness at the ITth point along the wedge
DFAC	Coefficient $\gamma^{-k^3/8} A^5 \epsilon$
DFAK	Quantity $DFAC \cdot t/(\tau N)^4$
EN	Parameter $N = M \sqrt{C/Re_t}$
EPS	Quantity $\epsilon = (\gamma - 1)/(\gamma + 1)$
EPS2	ϵ^2
ERFV	Quantity $1 + \text{erf}(S \sin \alpha)$ used in calculating free-molecule limit to heat flux
EXPV	Quantity $e^{-(S \sin \alpha)^2}$ in free-molecule heat flux calculations

FINAL	Logical variable, set to .TRUE. when calculations have been done for all of the specified distances from the leading edge for given angle of attack and nose radius
FL(K)	Array used by subroutine SHOCK to communicate its results to WEDGE. See glossary of symbols in SHOCK for definitions of the elements of FL
GAMMA	Specific-heat ratio, γ
GAMD	Double-precision value of GAMMA, for call to SHOCK routine
GFAC	Factor used in calculating Γ ; equal to $k/2A^2\epsilon M$ for the modified Cheng-Kemp theory, and to $\gamma k/2 A^2\epsilon$ for the unmodified theory
GFAK	$GFAC/(\tau N)^2$
GFC	Coefficient $k/2A^2\epsilon$ in GFAC
GMW	Mean molecular weight of gas, g/mole
G2	γ^2
G3	γ^3
HRATIO	Ratio h_w/h_0 of enthalpy at the wall to the free-stream stagnation enthalpy
HW	Gas enthalpy at the wall temperature, cal/g
H0	Free-stream stagnation enthalpy, cal/g
IA	Index of angles of attack
IAMAX	Index of angle of attack at which classical oblique shock solution gives a detached shock

IERR	Error indicator for classical oblique shock solution, set in SHOCK routine: 0 Detached shock 1 Attached shock - normal case 2 Convergence failure
IFM	Control variable; reset from 0 to 1 if any heat flux value is reduced to the free-molecule limit
IR	Index of leading edge radii
IS9	Absolute value of ISW9B
IT	Index of columns in the output table
ITAB	Index of blocks in the output table
ITH	Index equal to 1 for the modified Cheng-Kemp theory, equal to 2 for the unmodified theory
ITHS	Value of ITH before resetting because of IERR = 0
J	Implied DØ index in WRITE statements
NW1	Index of points in the uniform sequence of distances from the leading edge
NW2	Index of specified distances from the leading edge
ØMEGA	Parameter Ω , eq. I(502)
PFAC	Quantity $4 A^4 \epsilon^2 M^2 / \gamma k^2$
PIAK	Coefficient PFAC $\cdot (\tau N)^4$ in formula for pressure on wedge
PI	Value 3.1415927
PINF	Free-stream pressure, atm

PRATIØ	Ratio of pressure on wedge to free-stream pressure, p_w/p_1
PW:(IT)	Pressure on the wedge at the ITth distance from the leading edge, atm
QFM(IA)	Free-molecule limit to the heat flux for the IAth angle of attack, Btu/ft ² -sec
QFMC1	Coefficient $0.08885 p_{1a_e} \sqrt{RT_1/2 \pi W}$ in calculation of free-molecule heat flux, eq. I(512)
QFMC2	Coefficient $S^2 + [\gamma - \frac{1}{2} (\gamma+1) T_w/T_1]/(\gamma-1)$ in free-molecule heat flux calculation, eq. I(512)
QWW(IT)	Heat flux to the wedge at the ITth distance from the leading edge, Btu/ft ² -sec
REPF	Reynolds number per foot
RGC	Gas constant (8.314×10^7 erg/mole °K)
RHØINF	Free-stream density, ρ_1
RNAME	Hollerith routine name for call to subroutine DUMP
RU2	Dynamic pressure $\rho_1 u_1^2$
S	Free-molecule flow parameter $M \sqrt{\gamma/2}$
SQT	Quantity $\sqrt{zz' + \Omega \zeta}$ in modified Cheng-Kemp calculation; $\sqrt{zz'}$ in unmodified calculation
SQTPI	$\sqrt{\pi}$
SSA	$S \sin \alpha$
SSASQ	$(S \sin \alpha)^2$

TAU	Parameter $\tau = 0.664 + 1.73 h_w/h_0$
TAUN	τN
TAUN2	$(\tau N)^2$
TAUN3	$(\tau N)^3$
TAUN4	$(\tau N)^4$
TAUN6	$(\tau N)^6$
TINF	Free-stream temperature, T_1
TN	Diameter of leading edge, inch
TRATIØ	Ratio of wedge surface temperature to free-stream temperature
TREF	Reference temperature, $(1 + 3 h_w/h_0) T_0/6$, where T_0 is the free-stream stagnation temperature behind a frozen normal shock
UNIF	Free-stream velocity, u_1
UN(J)	Hollerith array for labelling the output as being based on the modified or unmodified Cheng-Kemp theory
VMUINF	Free-stream viscosity, lb/ft-sec
WK2	Square of leading-edge drag coefficient, k^2
WK3	k^3
WK4	k^4
WKV1	Next distance from the leading edge in the specified uniform sequence, inch

WXV2	Next separately specified distance from the leading edge, inch
XC	Coordinate x parallel to the free-stream flow for a point on the surface of the wedge at a distance XW(IT) from the leading edge; $XC = XW(IT) \cdot \cos \alpha$, inch
XM	Distance from the leading edge up to which merging effects are important (inch); see eq. I(510a)
XS	Distance from the leading edge beyond which the strong-interaction approximation breaks down (inch); see eq. I(510b)
XW(IT)	The ITth specified distance from the leading edge, inch
YS(IT)	Shock ordinate at the ITth distance from the leading edge, inch
YSFAC	Coefficient $\gamma^2 k^3 / 8 A^4 \epsilon$ in calculation of shock ordinate
YSFAK	Coefficient $YSFAC \cdot t / (\tau N)^4$ in calculation of shock ordinate
Z	Nondimensional shock ordinate z; eq. I(481a)
ZERØ	0.0
ZETA	Nondimensional coordinate ζ parallel to free-stream flow; see eq. I(481b)
ZFAC	Coefficient $16 A^7 \epsilon / \gamma^3 k^4$
ZFAK	Coefficient $ZFAC \cdot (\tau N)^6 / t$
ZTA(IT)	ζ at the ITth distance from the leading edge
ZZP	zz'
ZZPP	$(zz')'$

ZZPS	z..' as a single-precision value
4.135 <u>Subroutine WESØLN</u>	
B0	Coefficient $\sqrt[3]{6}$ in initial approximation to λ
B1	Coefficient $\sqrt{3}$ in initial approximation to λ
CALLED	Logical variable used to bypass resetting of the "C" coefficients after the first call to WESØLN
CAPGAM	Angle of attack parameter, Γ
CGSQ	Γ^2
C1	1. The "C" symbols are coefficients in the Cheng-Kemp analytical solution of the Cheng equation for $\Gamma = 0$
C2	2.
C3	1/3
C4	4.
C9	9.
C32	1.5
C103	10/3
C229	22/9
C463	46/3
C769	76/9
DLDZT	$d\lambda/d\zeta$ for Newton-Raphson iteration

N	Iteration counter
P1	$\zeta^{1/3}$
P2	$\zeta^{1/2}$
P12	$\zeta^{2/3}$
Q1	$1 + \sqrt{\lambda}$
Q2	$(1 + \sqrt{\lambda})^2$
Q3	$(1 + \sqrt{\lambda})^3$
Q4	$(1 + \sqrt{\lambda})^4$
Q5	$\ln(1 + \sqrt{\lambda})$
SQTL	$\sqrt{\lambda}$
XL	λ
Z	Nondimensional shock ordinate, z
ZETA	Nondimensional coordinate ζ parallel to the direction of free-stream flow
ZT	ζ calculated using trial value of λ
ZZP	zz'
ZZPP	$(zz')'$

4.136 Subroutine XSECT

Because a number of symbols are used for different purposes in different sections of subroutine XSECT, the symbols used in each section of the subroutine are defined separately below.

Statements to Statement 40

I(J)	Revised index for species J in master species list (common variable)
J	Index of species in master species list
JJ	Revised index for species not included in transport calculations
K	Count of species
L	Revised index for species included in transport calculations

Statements 40 to 110

J	Species index
K	Index of steps in the cross section calculation in the order in which they are carried out
KIJ	Index in packed lists of species pairs to which edited cross section data are applied
KM	Index of parameters for a single step of the cross section calculation
KV	Index in packed list of parameter values for all steps of the cross section calculation
L	Species index; step of cross section computations
LL	First index of species pair
LQ	NQ(L)
M	Index of the Kth step in the cross section calculation, equal to ISEQ(K)

MMV	Total number of parameters in V array
NK	Index of the option to be used in the Kth step; also, number of parameters used in step
<u>Statements 110 through 190</u>	
I(J)	Revised index for species J in master list (common variable)
II	First index of unused species pair
III	II-1
J	Second index of species pair referenced in V array; species index in master species list
JJ	Index of species added to cross section computation
K	Index in master species list of species added to cross section computations
L	Index of step in cross section computations
LL	First index of species pair referenced in V array
IQ	Location of species pair in IQ, JQ array
IQ1	Location in IQ, JQ array of first species pair for step
IQ2	Location in IQ, JQ array of last species pair for step
MV	Dummy variable (not used)

Statement 190 to Card XSE 213

IV	First index of species pair computed with option KQ = 10
J	Second index of species pair to be moved
JV	Second index of species pair computed with option KQ = 10
K	Count of array elements; dummy species index
L	Step in cross section computations; dummy species index
LL	First index of species pair to be moved
LQ	Location of species pair IV, JV in IQ, JQ array
LQ1	Location in IQ, JQ array of last species pair for step L-1 of computations
LQ2	Location in IQ, JQ array of last species pair for step L of computations
LV	Index of array elements to be moved
MM	Location in V array of last parameter for step L-1 of computations
MMV	Total number of parameters in V array
MV	Index for parameters in V array
NSV	Total number of steps in edited cross section computations

Card XSE 213 to Statement 300

II	First index of unused storage location for species pair
IV	First index of a species pair
J	Second index of species pair to be moved
JJ	Second index of unused storage location for species pair
JV	Second index of a species pair
L	Step of cross section computations
LL	First index of species pair to be moved
LQ	Index of species pair in IQ, JQ array
LQ2	Total number of species pairs in IQ, JQ array
MV	Dummy argument, value not required

Statement 380 to end of subroutine

I(K)	Species for which cross section data are to be supplied (common variable)
IE	For IE = 1, cross section data are not supplied for electrons

II	Absolute value of IZ for a species
IZ(L)	Charge of species L in units of electron charge (Note that IZ is negative for positively charged species)
J	Index indicating charge for species for which cross sections are to be supplied (see Table III)
JJ	Index indicating whether cross sections are to be supplied for electrons (see Table III)
KK	Index indicating charge of species for which cross sections are to be supplied (see Table III)
L	Step in cross section computations; species index
LL	Computational step at which additional cross section data are supplied for neutral-neutral interactions
NEWMV	Total number of parameters in V array
NN	Number of species for which cross section data are to be supplied

VCOUL(M,KK,J,JJ) Parameters V(M) for effective Coulomb cross sections between two species with the electronic charges KK and J, from eqs. I(100a). The parameters for JJ = 1 are for ion-ion collisions, and for JJ = 2 for ion-electron collisions

5. REFERENCES TO COMMON VARIABLES

This section presents a computer-generated analysis of references to common variables and common-equivalent variables in the NATA code. The variables are listed in alphanumeric order. For each variable, the name of the common block containing it is given. The block name / refers to unlabelled common. An asterisk following the block name indicates that the variable does not appear in common, but is equivalenced to a common variable in the designated block in one or more routines. The routines in which each variable is referenced are indicated by numbers, based on the list of routine names given on the first page. Routine number 2 is the block data routine containing species, reaction, geometry, and other data; number 3 is the transport cross section block data routine. The routines EXP and RESET are missing from the list because the analysis was carried out for the UNIVAC version of NATA, which does not contain these routines. Neither EXP nor RESET contains any common blocks.

Lines containing a blank variable name are continuations of the preceding lines, in cases in which the number of referencing routines is too large to be printed on a single line.

Appearance of a variable in a common or equivalence statement is not considered to constitute a reference. A routine is listed as referencing a given variable if the variable appears in an arithmetic or DATA statement, a READ or WRITE statement or a NAMELIST statement, or is used as an index or a subroutine or function argument.

In the case of a common-equivalent variable, the program used to prepare the list checks for the presence of the associated common block in the routine, but does not verify the presence of the equivalence statement in the routine. For this reason, some of the listed references to common-equivalent variables are spurious. For example, in subroutine BLAYER, the variable X is equivalenced to the

variable CX in unlabelled common. Subroutines GEØMAR, ØUT1, and TRANSP are also listed as referencing X. All three of these references are spurious because X is not equivalenced to a common variable in these three routines.

In some cases, different names are used for the same common variable in different routines. All such names are included in the list below, each with its own list of referencing routines.

INDICES OF ROUTINES

1	MAIN	36	NRMAX
2	BLOCKD	37	OUT
3	BLOCKD	38	DUT1
4	AGSOLN	39	PERT
5	AGSOLN	40	PIOMEG
6	AXFIT	41	PRQP
7	AXSFACT	42	PRTA
8	RLAYER	43	PUTQIN
9	RLCALL	44	QCJUL
10	BXSFACT	45	QFX
11	COMM	46	QEXP
12	CXSFACT	47	QINTRP
13	DEPIVS	48	OLJ
14	DSMSOL	49	OMIX
15	DUMPEX	50	QREPP
16	ELCOND	51	OSAME
17	ELTIME	52	OTAB
18	EPART	53	Q11
19	FOCALC	54	Q12
20	EXACT	55	Q13
21	FINDX	56	Q14
22	FR07FQ	57	RADIUS
23	GEOM	58	READ
24	GEOMAR	59	RESTMP
25	INIT	60	RNKT
26	KANDMU	61	SHOCK
27	KINT	62	SIMQ
28	LIST	63	STUNTS
29	MATINV	64	THFRM
30	MODEL	65	THROAT
31	NEWRAP	66	TRANSP
32	NEXTMP	67	WEDGE
33	NONEC	68	WFSOLN
34		69	XSFCT
35			

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

COMMON

VARIABLE

AA	1. 16. 24.	20. 33. 35. 39.	
AAA	21. 58.		
AAMS			
ACOM			
AE			
AENTS	1. 16. 35. 42.	11. 16. 23. 24. 35. 38. 60. 65.	
AFNX	12. 16. 26. 35.		
AIN			
ALGJ			
AM			
AMACH	16. 21. 23. 32. 39. 42. 65.		
AMP	18. 16.		
AMPP	2. 30. 58. 67.		
ANGLE	16. 27. 35.		
AR	16. 27. 35.		
ARHA	16. 27. 35.		
ARRB	50. 58.		
ASTAR	16. 25. 58.		
ATPI	2. 58.		
AXIMOD	23. 30. 32. 35. 58.		
AXISYM			
AZ	28. 29. 66.		
B	16. 39. 39.		
BETA	11. 21. 26. 33.		
BLK	11. 21. 35.		
BLINT	18. 16. 35.		
BLNEAR	35.		
BLVAR	23. 11.		
BPAR	11. 16. 58.		
BRSTAR	28. 58.		
BTA	16. 20. 33. 41. 58.		
BZERO	2. 58.		
BZEROI	1. 16. 24. 66.		
C	11. 16. 30. 35. 58.		
CAI	26. 30. 58.		
CAPD	16. 26. 33. 36. 42.		
CAPX	16. 36.		
CAPXTH	16. 35.		
CAR3	18. 23. 42.		
CAS	2. 16. 30. 35. 58.		
CATFAC	2. 30. 35. 58.		
CCHI	26. 8. 11. 13. 16. 21. 38. 39. 63. 64. 66.		
CCI	1. 21. 26. 33. 35.		
CCPJ	20. 16. 30. 58.		
CDIJ			
CEACT			

ORIGINAL PAGE IS
OF POOR QUALITY

VARIABLE

COMMON

REFERENCING ROUTINES

CF	/CONVRT/	1. 16. 32. 38.	
CGI		11. 16. 20. 26. 33. 66.	
CGMU		16. 26. 33.	
CGMW	/COLDSP/	1. 8. 59.	
CH		1. 11. 16. 23. 26. 32. 33. 38. 41. 59. 60.	
CHA		1. 8. 11. 16. 23. 26. 32. 33. 35. 38. 41. 59. 60.	
CHANAM	/RDOUT	38. 58.	
CHB	/TNERK	35. 60. 38. 42.	
CHI		11. 16. 33.	
CHI1		16. 20. 33.	
CLNPI		11. 16. 64.	
CLNT		13. 16. 64.	
CLNTD	/TNDC	11. 13. 16.	
CLNIMC		11. 16.	
CM		1. 8. 9. 11. 16. 23. 26. 32. 33. 35. 38. 39. 41.	
CMA		1. 8. 11. 16. 21. 23. 26. 32. 33. 35. 38. 39. 41.	
CMW		59. 63. 30. 58.	
CRUPLD	/BINE	16. 26. 30. 58.	
CP	/CHAN	58. 16. 35.	
CPWALL	/DL	1. 8. 16. 32. 39. 67.	
CP1	/CHAN	1. 2.	
CP2	/CHAN		
CP3	/CHAN		
CP4	/CHAN		
CP5	/CHAN		
CRA			
CRP			
CRB			
CRS			
CSTA			
CT			
CTAP			
CTAPI	/READAT	1. 11. 16. 26. 27. 32. 63.	
CTB		1. 8. 11. 16. 21. 27. 32. 35. 38. 59. 63.	
CIC		16. 23. 26. 33.	
CTD		11. 16. 35.	
CTF	/TNDC	9. 11. 13. 16. 20. 21. 23. 27. 32. 33.	
CTF3	/TNONEO	35. 36. 38. 39. 41. 42. 63. 64. 66.	
CTMAX	/TNERK	1. 8. 11. 16. 20. 21. 23. 26. 27. 32. 35. 38. 39.	
CTMXX		58. 59. 63. 64. 66.	
CTMXXI	/MASSFL	58. 60.	
CTP		16. 35. 40.	
CTPL		16. 16. 21. 27. 35. 38. 42. 60.	
CTT		11. 13. 16. 21. 27. 35. 38. 42. 60.	
CX		11. 13. 16. 21. 27. 35. 38. 42. 60.	
CXB		16. 27. 64.	
CXMAX		16. 27. 64.	
CXMAXI		16. 27. 64.	
CZ	/READAT	1. 16. 35. 42.	
DATAP		1. 16. 35. 42.	
DATAPUT		1. 16. 35. 42.	

ORIGINAL PAGE IS
OF POOR QUALITY

VARIABLE

COMMON

REFERENCING ROUTINE

DATEST	/	16.	27.	35.	
DBTEST	/	16.	27.		
DCA	/	35.			
DCHA	/	8.	21.	60.	
DCHLL	/	2.	35.	58.	
DCHPAT	/	2.	35.	58.	
DCI	/	35.			
COBLP	/	8.	16.		
COELBL	/	4.	8.	16.	35.
DELRL	/	1.	4.	6.	8.
DELBLP	/	8.	16.		16.
DELTAX	/	16.	35.	58.	60.
DELTXI	/	2.	58.		
DELT1	/	1.	16.	23.	35.
DELT11	/	2.	58.		58.
DELT2	/	16.			
DGJ	/	21.	38.	39.	60.
DH	/	29.	66.		
DIAM	/	1.	16.	38.	58.
DLGOX	/	38.			
DLGGA	/	11.	13.	14.	21.
DLOGR	/	11.	16.	21.	24.
DOMAX	/	35.	60.	35.	42.
DT	/	8.	16.	21.	42.
DTE	/	58.	16.	21.	60.
DUM	/	2.	30.	58.	67.
DWX	/				
DXMDDP	/				
DXMP	/				
DXP	/	8.	23.	35.	
DXP3	/				
EEP	/	58.			
EEP1	/	58.			
EEP10	/	58.			
EEP2	/	58.			
EEP3	/	58.			
EEP4	/	58.			
EEP5	/	58.			
EEP6	/	58.			
EEP7	/	58.			
EEP8	/	58.			
EEP9	/	58.			
ELJ	/	27.	30.	58.	64.
ELMENT	/	30.	58.		
ENT	/	16.	23.	41.	58.
EPAR	/	16.	19.	30.	58.
EPAP	/	58.			
EPSE	/	16.	21.	32.	35.
EPSLON	/	16.	23.	32.	35.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCE ROUTINES

ROUTINE	COMMON	REFERENCE ROUTINES
EP1	/ELEM	2.
EP10	/ELEM	
EP2	/ELEM	2.
EP3	/ELEM	2.
EP4	/ELEM	2.
EP5	/ELEM	2.
EP6	/ELEM	2.
EP7	/ELEM	
EP8	/ELEM	
EP9	/ELEM	
ERR	/ERROR	1. 8. 11. 13. 15. 20. 21. 22. 23. 24. 26.
ETAJ	/ERROR	32. 33. 34. 35. 36. 38. 39. 42. 59. 60. 65. 66. 69.
ETACJ	/	11. 16. 27. 30. 58. 64.
FACNAM	/NEWMP	16. 34. 58.
FAILED	/RDOUT	38. 58.
FLEWIS	/CERROR	11. 13. 16. 35. 60.
FLEWR	/TRPROR	1. 16. 32. 66.
FLOW	/RESRPR	1.
FLUX	/RDOUT	1. 38. 58.
FSTAG	/	1. 16. 23. 33. 35. 36. 41.
FVOUT	/MODPAR	2. 16. 30. 32. 58.
GAMIN	/OUTPUT	38. 47. 58.
GAMMA	/RENE	2. 35. 58.
GELJ	/OUTPUT*	67. 58.
GJ	/	30. 58. 64.
GJA	/	1. 8. 9. 11. 21. 23. 32. 33. 35. 36. 38. 39. 60.
GJE	/	1. 16. 23. 26. 35. 36. 41. 63.
GJMF	/	16. 35. 60.
GMA	/OUTPUT	67.
GMRP	/OUTPUT*	58.
GP1	/MIXT	2.
GP2	/MIXT	2.
GP3	/MIXT	2.
GP4	/MIXT	2.
GP5	/MIXT	2.
GP6	/MIXT	2.
GSO	/MIXT	2.
GTFST	/RENE	35. 58.
H	/	2. 38. 59.
HDFLX	/	8. 35. 60.
HR	/	16. 32. 35. 38. 58. 63.
HS	/BLOUT	2. 8. 16. 38.
HSTAG	/ROMAIN	32. 38. 58. 59.
HTEST	/READAT	58.
IAMBIP	/RENE	2. 35. 63. 69.
IATCM	/TRANS2	7. 1. 8. 36.
IC	/RDMCD	30. 69.
	/LN	16. 32. 33. 38. 58.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

VARIABLE	COMMON	REFERENCING ROUTINES
ICASE	/READAT/	2. 58. 58.
ICCH	/TNONEO/	11. 16. 58.
ICHAN	/READAT/	2. 58.
ICHAND	/READAT/	2. 58.
ICI	/RDLIST/	58. 69.
ID1	/READAT/	66. 69.
ID2	/TRANSE/	66. 69.
IF	/READAT/	58.
IFEE	/READAT/	58.
IELEC	/TRANSE/	17. 69.
IFAIL	/TNERK/	35. 60.
IFLOW	/TAPOUT/	
IGAS	/RDLIST/	2. 58.
IGASO	/RDLIST/	2. 58.
IGJ	/	16. 27. 30. 58. 63. 64.
IGM	/	16. 27. 30. 58. 64.
II111	/TRANS2/	
II110	/TRANI/	63. 69.
II1100	/TRANI/	3. 69.
II111	/TRANI/	3.
II112	/TRANI/	3.
II113	/TRANI/	3.
II114	/TRANI/	3.
II115	/TRANI/	3.
II117	/TRANI/	3.
II119	/TRANI/	3.
II12	/TRANI/	3.
II120	/TRANI/	3.
II121	/TRANI/	3.
II122	/TRANI/	3.
II123	/TRANI/	3.
II124	/TRANI/	3.
II125	/TRANI/	3.
II126	/TRANI/	3.
II127	/TRANI/	3.
II128	/TRANI/	3.
II129	/TRANI/	3.
II13	/TRANI/	3.
II130	/TRANI/	3.
II131	/TRANI/	3.
II132	/TRANI/	3.
II133	/TRANI/	3.
II134	/TRANI/	3.
II135	/TRANI/	3.
II136	/TRANI/	3.
II137	/TRANI/	3.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINE

1138	/	3.
1139	/	3.
1140	/	3.
1141	/	3.
1142	/	3.
1143	/	3.
1144	/	3.
1145	/	3.
1146	/	3.
1147	/	3.
1148	/	3.
1149	/	3.
1150	/	3.
1151	/	3.
1152	/	3.
1153	/	3.
1154	/	3.
1155	/	3.
1156	/	3.
1157	/	3.
1158	/	3.
1159	/	3.
1160	/	3.
1161	/	3.
1162	/	3.
1163	/	3.
1164	/	3.
1165	/	3.
1166	/	3.
1167	/	3.
1168	/	3.
1169	/	3.
1170	/	3.
1171	/	3.
1172	/	3.
1173	/	3.
1174	/	3.
1175	/	3.
1176	/	3.
1177	/	3.
1178	/	3.
1179	/	3.
1180	/	3.
1181	/	3.
1182	/	3.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCE ROUTINES

COMMON

TABLE

ISSP3	/	16.	21.	35.	39.	60.
ISSP4	/	16.	21.	35.	60.	
ISUPR	/	35.				
ISW1A	/	1.	16.	58.		
ISW1B	/	1.	16.	58.	63.	66.
ISW2A	/	1.	16.	58.		
ISW2B	/	1.	16.	58.	59.	59.
ISW3A	/	1.	16.	58.		
ISW3B	/	1.	16.	9.	13.	16.
ISW4A	/	1.	16.	58.	23.	24.
ISW4B	/	1.	16.	58.	30.	35.
ISW5A	/	1.	16.	58.	38.	42.
ISW5B	/	1.	16.	58.	58.	65.
ISW6A	/	1.	16.	58.		
ISW6B	/	1.	16.	58.		
ISW7B	/	2.	58.	47.	58.	66.
ISW9B	/	2.	58.	67.		
ITB	/	16.	35.			
ITPOUT	/	1.	2.	16.	32.	38.
ITR	/	16.	19.	30.	58.	
ITYPER	/	1.	32.	38.		
IUPD	/	1.	16.	21.	35.	42.
IZERO	/	1.	16.	30.	58.	64.
I1	/	58.				
I1C	/	58.				
I1CO	/	58.				
I11	/	58.				
I12	/	58.				
I13	/	58.				
I14	/	58.				
I15	/	58.				
I16	/	58.				
I17	/	58.				
I18	/	58.				
I19	/	58.				
I2	/	58.				
I20	/	58.				
I21	/	58.				
I22	/	58.				
I23	/	58.				
I24	/	58.				
I25	/	58.				
I26	/	58.				
I27	/	58.				
I28	/	58.				
I29	/	58.				
I3	/	58.				
I30	/	58.				
I31	/	58.				

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

COMMON

VARIABLE

132	/TRANI	58.
133	/TRANI	58.
134	/TRANI	58.
135	/TRANI	58.
136	/TRANI	58.
137	/TRANI	58.
138	/TRANI	58.
139	/TRANI	58.
140	/TRANI	58.
141	/TRANI	58.
142	/TRANI	58.
143	/TRANI	58.
144	/TRANI	58.
145	/TRANI	58.
146	/TRANI	58.
147	/TRANI	58.
148	/TRANI	58.
149	/TRANI	58.
150	/TRANI	58.
151	/TRANI	58.
152	/TRANI	58.
153	/TRANI	58.
154	/TRANI	58.
155	/TRANI	58.
156	/TRANI	58.
157	/TRANI	58.
158	/TRANI	58.
159	/TRANI	58.
160	/TRANI	58.
161	/TRANI	58.
162	/TRANI	58.
163	/TRANI	58.
164	/TRANI	58.
165	/TRANI	58.
166	/TRANI	58.
167	/TRANI	58.
168	/TRANI	58.
169	/TRANI	58.
170	/TRANI	58.
171	/TRANI	58.
172	/TRANI	58.
173	/TRANI	58.
174	/TRANI	58.
175	/TRANI	58.
176	/TRANI	58.
177	/TRANI	58.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

VARIABLE

COMMON

178	/TRANI	58.
179	/TRANI	58.
180	/TRANI	58.
181	/TRANI	58.
182	/TRANI	58.
183	/TRANI	58.
184	/TRANI	58.
185	/TRANI	58.
186	/TRANI	58.
187	/TRANI	58.
188	/TRANI	58.
189	/TRANI	58.
190	/TRANI	58.
191	/TRANI	58.
192	/TRANI	58.
193	/TRANI	58.
194	/TRANI	58.
195	/TRANI	58.
196	/TRANI	58.
197	/TRANI	58.
198	/TRANI	58.
199	/TRANI	58.
JATOM	/LN	58.
JCS	/READAT	58.
JDIM	/BL	1. 2. 4. 5. 8. 16. 24. 25. 38. 58.
JJK		1. 16. 27. 33.
JJO		63. 69.
JJ1	/TRANJ	3.
JJ10	/TRANJ	3.
JJ100	/TRANJ	3.
JJ11	/TRANJ	3.
JJ12	/TRANJ	3.
JJ13	/TRANJ	3.
JJ14	/TRANJ	3.
JJ15	/TRANJ	3.
JJ16	/TRANJ	3.
JJ17	/TRANJ	3.
JJ18	/TRANJ	3.
JJ19	/TRANJ	3.
JJ2	/TRANJ	3.
JJ20	/TRANJ	3.
JJ21	/TRANJ	3.
JJ22	/TRANJ	3.
JJ23	/TRANJ	3.
JJ24	/TRANJ	3.
JJ25	/TRANJ	3.
JJ26	/TRANJ	3.
JJ27	/TRANJ	3.

REFERENCING ROUTINES

COMMON

VARIABLE

JJ28	/TRANJ /	3.
JJ29	/TRANJ /	3.
JJ3	/TRANJ /	3.
JJ30	/TRANJ /	3.
JJ31	/TRANJ /	3.
JJ32	/TRANJ /	3.
JJ33	/TRANJ /	3.
JJ34	/TRANJ /	3.
JJ35	/TRANJ /	3.
JJ36	/TRANJ /	3.
JJ37	/TRANJ /	3.
JJ38	/TRANJ /	3.
JJ39	/TRANJ /	3.
JJ4	/TRANJ /	3.
JJ40	/TRANJ /	3.
JJ41	/TRANJ /	3.
JJ42	/TRANJ /	3.
JJ43	/TRANJ /	3.
JJ44	/TRANJ /	3.
JJ45	/TRANJ /	3.
JJ46	/TRANJ /	3.
JJ47	/TRANJ /	3.
JJ48	/TRANJ /	3.
JJ49	/TRANJ /	3.
JJ5	/TRANJ /	3.
JJ50	/TRANJ /	3.
JJ51	/TRANJ /	3.
JJ52	/TRANJ /	3.
JJ53	/TRANJ /	3.
JJ54	/TRANJ /	3.
JJ55	/TRANJ /	3.
JJ56	/TRANJ /	3.
JJ57	/TRANJ /	3.
JJ58	/TRANJ /	3.
JJ59	/TRANJ /	3.
JJ6	/TRANJ /	3.
JJ60	/TRANJ /	3.
JJ61	/TRANJ /	3.
JJ62	/TRANJ /	3.
JJ63	/TRANJ /	3.
JJ64	/TRANJ /	3.
JJ65	/TRANJ /	3.
JJ66	/TRANJ /	3.
JJ67	/TRANJ /	3.
JJ68	/TRANJ /	3.
JJ69	/TRANJ /	3.
JJ7	/TRANJ /	3.
JJ70	/TRANJ /	3.
JJ71	/TRANJ /	3.
JJ72	/TRANJ /	3.

VARIABLE

-359-

VARIABLE COMMON REFERENCING ROUTINES

J26	/TRANJ /	59.
J27	/TRANJ /	58.
J28	/TRANJ /	58.
J29	/TRANJ /	58.
J3	/TRANJ /	58.
J30	/TRANJ /	58.
J31	/TRANJ /	58.
J32	/TRANJ /	58.
J33	/TRANJ /	58.
J34	/TRANJ /	58.
J35	/TRANJ /	58.
J36	/TRANJ /	58.
J37	/TRANJ /	58.
J38	/TRANJ /	58.
J39	/TRANJ /	58.
J4	/TRANJ /	58.
J40	/TRANJ /	58.
J41	/TRANJ /	58.
J42	/TRANJ /	58.
J43	/TRANJ /	58.
J44	/TRANJ /	58.
J45	/TRANJ /	58.
J46	/TRANJ /	58.
J47	/TRANJ /	58.
J48	/TRANJ /	58.
J49	/TRANJ /	58.
J5	/TRANJ /	58.
J50	/TRANJ /	58.
J51	/TRANJ /	58.
J52	/TRANJ /	58.
J53	/TRANJ /	58.
J54	/TRANJ /	58.
J55	/TRANJ /	58.
J56	/TRANJ /	58.
J57	/TRANJ /	58.
J58	/TRANJ /	58.
J59	/TRANJ /	58.
J6	/TRANJ /	58.
J60	/TRANJ /	58.
J61	/TRANJ /	58.
J62	/TRANJ /	58.
J63	/TRANJ /	58.
J64	/TRANJ /	58.
J65	/TRANJ /	58.
J66	/TRANJ /	58.
J67	/TRANJ /	58.
J68	/TRANJ /	58.
J69	/TRANJ /	58.
J7	/TRANJ /	58.
J70	/TRANJ /	58.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

COMMON

VARIABLE

PPESAI	/READAT/	5P.	23.	23.	36.	
PPESB	/	16.	16.	23.	36.	
PRESTH	/	66.	16.	35.	39.	42.
PRF	/TRPROP/	9.	16.	35.	39.	42.
PRHD	/	1.	8.	16.	38.	
PRR	/RESPRP/	2.	17.	28.	29.	43.
PRREF	/BLOUT/	7.	16.	21.	35.	38.
Q	/TRANS1/	11.	16.	21.	38.	
QDPC	/TNCE/	11.	16.	21.	38.	
QDPR	/TNCE/	11.	16.	21.	38.	
QEVAX	/TNERK/	16.	20.	26.	33.	
QV	/	16.	1.	8.	32.	58.
QVJ	/COLDSP/	11.	16.	30.	58.	
QJ	/	12.	35.	58.		
QJEST	/RENE/	2.	8.	16.	38.	
QWDOT	/BLOUT/	2.	30.	55.	67.	
QADLE	/RDWEDG/	2.	58.			
READG	/READAT/	2.	58.			
READXS	/READAT/	2.	58.			
REF	/BLOUT/	2.	8.	16.	38.	
RH	/	8.	11.	16.	21.	26.
RHAP	/	1.	9.	11.	16.	21.
RHC	/	41.	42.	60.		
RHOBAR	/	16.	23.	35.		
RHOC	/	16.	23.	33.	41.	
RHOCINF	/	16.				
RHOC2	/OUTPUT/	11.	16.	21.		
RHPL	/	11.	16.	23.	24.	36.
RHPTH	/	11.	16.	23.	26.	33.
RHOPARA	/	11.	16.	23.	33.	41.
RHOPARP	/	16.	26.	33.	41.	
RPRD	/	58.				
RP1	/REAC	2.	58.			
RP17	/REAC	2.	58.			
RP11	/REAC	2.	58.			
RP12	/REAC	2.	58.			
RP13	/REAC	2.	58.			
RP14	/REAC	2.	58.			
RP15	/REAC	2.	58.			
RP16	/REAC	2.	58.			
RP17	/REAC	2.	58.			
RP18	/REAC	2.	58.			
RP19	/REAC	2.	58.			
RP20	/REAC	2.	58.			
RP21	/REAC	2.	58.			
RP22	/REAC	2.	58.			
RP23	/REAC	2.	58.			
RP24	/REAC	2.	58.			

TABLE REFERENCING ROUTINES

RP25	/REAC	/	2.	58.
RP26	/REAC	/	2.	58.
RP27	/REAC	/	58.	58.
RP28	/REAC	/	58.	58.
RP29	/REAC	/	58.	58.
RP30	/REAC	/	2.	58.
RP31	/REAC	/	58.	58.
RP32	/REAC	/	58.	58.
RP33	/REAC	/	58.	58.
RP34	/REAC	/	58.	58.
RP35	/REAC	/	2.	58.
RP36	/REAC	/	2.	58.
RP37	/REAC	/	2.	58.
RP38	/REAC	/	2.	58.
RP39	/REAC	/	2.	58.
RP40	/REAC	/	2.	58.
RP41	/REAC	/	2.	58.
RP42	/REAC	/	2.	58.
RP43	/REAC	/	2.	58.
RP44	/REAC	/	2.	58.
RP45	/REAC	/	2.	58.
RP46	/REAC	/	2.	58.
RP47	/REAC	/	2.	58.
RP48	/REAC	/	2.	58.
RP49	/REAC	/	2.	58.
RP50	/REAC	/	2.	58.
RP51	/REAC	/	2.	58.
RP52	/REAC	/	2.	58.
RP53	/REAC	/	2.	58.
RP54	/REAC	/	2.	58.
RP55	/REAC	/	2.	58.
RP56	/REAC	/	2.	58.
RP57	/REAC	/	2.	58.
RP58	/REAC	/	2.	58.
RP59	/REAC	/	2.	58.
RP60	/REAC	/	2.	58.
RP61	/REAC	/	2.	58.
RP62	/REAC	/	2.	58.
RP63	/REAC	/	2.	58.
RP64	/REAC	/	2.	58.
RP65	/REAC	/	2.	58.
RP66	/REAC	/	2.	58.
RP67	/REAC	/	2.	58.
RP68	/REAC	/	2.	58.
RP69	/REAC	/	2.	58.
RP70	/REAC	/	2.	58.

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

REFERENCING ROUTINES

COMMON

VARIABLE

T97	/OUTMOD/	32	38		
TEST	/	16	20	27	33
TESTA	/	16	23	27	36
TETEST	/RENE	2	35	58	
TFA	/	16	30	58	64
TFR	/	16	27	30	58
TFC	/	16	27	30	58
TFO	/	16	27	30	58
TFE	/	16	27	30	58
TFK	/	16	30	58	64
THETA	/BLOUT	2	8	16	38
THEV	/	16	27	64	
THEVE	/POLYAT	27	30	58	64
THEVP	/	30	58		
TINF	/OUTPUT/*	67	58		
TL	/TRANS4/	47	58		
TL1ST	/TNEQ	40	58		
TL1	/TRANS4/	3			
TL10	/TRANS4/	3			
TL11	/TRANS4/	3			
TL12	/TRANS4/	3			
TL13	/TRANS4/	3			
TL14	/TRANS4/	3			
TL15	/TRANS4/	3			
TL16	/TRANS4/	3			
TL17	/TRANS4/	3			
TL18	/TRANS4/	3			
TL19	/TRANS4/	3			
TL2	/TRANS4/	3			
TL20	/TRANS4/	3			
TL21	/TRANS4/	3			
TL22	/TRANS4/	3			
TL23	/TRANS4/	3			
TL24	/TRANS4/	3			
TL25	/TRANS4/	3			
TL26	/TRANS4/	3			
TL27	/TRANS4/	3			
TL28	/TRANS4/	3			
TL29	/TRANS4/	3			
TL3	/TRANS4/	3			
TL30	/TRANS4/	3			
TL31	/TRANS4/	3			
TL32	/TRANS4/	3			
TL33	/TRANS4/	3			
TL34	/TRANS4/	3			
TL35	/TRANS4/	3			
TL36	/TRANS4/	3			
TL37	/TRANS4/	3			
TL38	/TRANS4/	3			
TL39	/TRANS4/	3			

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCING ROUTINES

VARIABLE

COMMON

TL4	/TRANS4/	3.
TL40	/TRANS4/	3.
TL5	/TRANS4/	3.
TL6	/TRANS4/	3.
TL7	/TRANS4/	3.
TL8	/TRANS4/	3.
TL9	/TRANS4/	3.
TMODEL	/MODPAR/	2. 16. 30. 32. 59.
TNEP	/TNE/	58.
TN1	/TNE/	2.
TN2	/TNE/	2.
TPLATF	/MODPAR/	2. 30. 58.
TPRINT	/	16. 35. 58.
TPRVTI	/READAT/	2. 58.
TPROP	/	62.
TR0	/TPPROP/	8.
TS	/STAG/	16. 32.
TSAR	/MODPT/	16. 34. 58.
TSDIAM	/MODPT/	16. 30. 58.
TSF	/STAG/	16. 23. 27. 35. 58.
TSTOP	/	16. 23. 27. 58.
TSTOPT	/MASSFL/	2. 27. 58.
TT	/	63.
TTST	/RENE/	2. 35. 58.
TWALL	/BL/	1. 2. 8. 16. 38. 58.
TWEDGE	/ROWEDG/	1. 30. 58. 67.
T20	/STAG/	23. 32. 35.
T2F.	/STAG/	23. 32. 35.
UINF	/OUTPUT/	8.
UP	/	67.
US	/SS/	16. 35.
V	/TRANS7/	1. 2. 21. 23. 42.
VISC	/TPPROP/	10. 43. 63. 69.
VISCROT	/RESPRP/	1. 8. 16. 23. 32. 39. 66. 67.
VMUINF	/BL/	1. 23.
VV	/OUTPUT/	1. 16.
VV1	/TRANV/	63. 69.
VV10	/TRANV/	3.
VV100	/TRANV/	3.
VV11	/TRANV/	3.
VV12	/TRANV/	3.
VV13	/TRANV/	3.
VV14	/TRANV/	3.
VV15	/TRANV/	3.
VV16	/TRANV/	3.
VV17	/TRANV/	3.
VV18	/TRANV/	3.
VV19	/TRANV/	3.

REFERENCE ROUTINES

VARIABLE

COMMON

VV2	/TRANV /	3.
VV20	/TRANV /	3.
VV21	/TRANV /	3.
VV22	/TRANV /	3.
VV23	/TRANV /	3.
VV24	/TRANV /	3.
VV25	/TRANV /	3.
VV26	/TRANV /	3.
VV27	/TRANV /	3.
VV28	/TRANV /	3.
VV29	/TRANV /	3.
VV3	/TRANV /	3.
VV10	/TRANV /	3.
VV11	/TRANV /	3.
VV32	/TRANV /	3.
VV33	/TRANV /	3.
VV34	/TRANV /	3.
VV35	/TRANV /	3.
VV36	/TRANV /	3.
VV37	/TRANV /	3.
VV38	/TRANV /	3.
VV39	/TRANV /	3.
VV4	/TRANV /	3.
VV40	/TRANV /	3.
VV41	/TRANV /	3.
VV42	/TRANV /	3.
VV43	/TRANV /	3.
VV44	/TRANV /	3.
VV45	/TRANV /	3.
VV46	/TRANV /	3.
VV47	/TRANV /	3.
VV48	/TRANV /	3.
VV49	/TRANV /	3.
VV5	/TRANV /	3.
VV50	/TRANV /	3.
VV51	/TRANV /	3.
VV52	/TRANV /	3.
VV53	/TRANV /	3.
VV54	/TRANV /	3.
VV55	/TRANV /	3.
VV56	/TRANV /	3.
VV57	/TRANV /	3.
VV58	/TRANV /	3.
VV59	/TRANV /	3.
VV6	/TRANV /	3.
VV60	/TRANV /	3.
VV61	/TRANV /	3.
VV62	/TRANV /	3.
VV63	/TRANV /	3.
VV64	/TRANV /	3.

REFERENCING ROUTINES

RIABLE COMMON

VV65	/TRANV /	
VV66	/TRANV /	
VV67	/TRANV /	
VV68	/TRANV /	
VV69	/TRANV /	
VV7	/TRANV /	
VV70	/TRANV /	
VV71	/TRANV /	
VV72	/TRANV /	
VV73	/TRANV /	
VV74	/TRANV /	
VV75	/TRANV /	
VV76	/TRANV /	
VV77	/TRANV /	
VV78	/TRANV /	
VV79	/TRANV /	
VV8	/TRANV /	
VV80	/TRANV /	
VV81	/TRANV /	
VV82	/TRANV /	
VV83	/TRANV /	
VV84	/TRANV /	
VV85	/TRANV /	
VV86	/TRANV /	
VV87	/TRANV /	
VV88	/TRANV /	
VV89	/TRANV /	
VV9	/TRANV /	
VV90	/TRANV /	
VV91	/TRANV /	
VV92	/TRANV /	
VV93	/TRANV /	
VV94	/TRANV /	
VV95	/TRANV /	
VV96	/TRANV /	
VV97	/TRANV /	
VV98	/TRANV /	
VV99	/TRANV /	
V1	/TRANV /	58.
V10	/TRANV /	58.
V100	/TRANV /	58.
V11	/TRANV /	58.
V12	/TRANV /	58.
V13	/TRANV /	58.
V14	/TRANV /	58.
V15	/TRANV /	58.
V16	/TRANV /	58.
V17	/TRANV /	58.
V18	/TRANV /	58.
V19	/TRANV /	58.

3.

3.

3.

REFERENCING ROUTINES

VARIABLE	COMMON	
V2	/TRANV /	58.
V20	/TRANV /	58.
V21	/TRANV /	58.
V22	/TRANV /	58.
V23	/TRANV /	58.
V24	/TRANV /	58.
V25	/TRANV /	58.
V26	/TRANV /	58.
V27	/TRANV /	58.
V28	/TRANV /	58.
V29	/TRANV /	58.
V3	/TRANV /	58.
V30	/TRANV /	58.
V31	/TRANV /	58.
V32	/TRANV /	58.
V33	/TRANV /	58.
V34	/TRANV /	58.
V35	/TRANV /	58.
V36	/TRANV /	58.
V37	/TRANV /	58.
V38	/TRANV /	58.
V39	/TRANV /	58.
V4	/TRANV /	58.
V40	/TRANV /	58.
V41	/TRANV /	58.
V42	/TRANV /	58.
V43	/TRANV /	58.
V44	/TRANV /	58.
V45	/TRANV /	58.
V46	/TRANV /	58.
V47	/TRANV /	58.
V48	/TRANV /	58.
V49	/TRANV /	58.
V5	/TRANV /	58.
V50	/TRANV /	58.
V51	/TRANV /	58.
V52	/TRANV /	58.
V53	/TRANV /	58.
V54	/TRANV /	58.
V55	/TRANV /	58.
V56	/TRANV /	58.
V57	/TRANV /	58.
V58	/TRANV /	58.
V59	/TRANV /	58.
V6	/TRANV /	58.
V60	/TRANV /	58.
V61	/TRANV /	58.
V62	/TRANV /	58.
V63	/TRANV /	58.
V64	/TRANV /	58.

REFERENCING ROUTINES

COMMON

TABLE

-372-

REFERENCING ROUTINES

COMMON

VARIABLE

XNUI	/	11.	16.	35.
XNUIJ	/	11.	30.	35.
XNUIJP	/	11.	30.	58.
XP	/	11.	30.	58.
XPB	/	18.	23.	35.
XSN	/	16.	23.	35.
XXX	/	2.	8.	15.
XZERO	/	1.	38.	35.
XZEROI	/	8.	16.	23.
YOZO	/	58.	16.	58.
ZCAP	/	9.	57.	
ZCH	/	16.	20.	26.
ZCM	/	16.	20.	32.
ZGMU	/	16.	20.	32.
ZH2	/	20.	26.	
ZH2	/	28.	65.	69.
ZPA	/	16.	33.	41.
ZPRD	/	16.	23.	26.
ZPZ	/	58.	23.	33.
ZP1	/	20.	26.	
ZP10	/	2.		
ZP11	/	2.		
ZP12	/	2.		
ZP13	/	2.		
ZP14	/	2.		
ZP15	/			
ZP16	/			
ZP17	/			
ZP18	/			
ZP19	/			
ZP2	/	2.		
ZP30	/			
ZP3	/	2.		
ZP4	/	2.		
ZP5	/	2.		
ZP6	/	2.		
ZP7	/	2.		
ZP8	/	2.		
ZP9	/	2.		
ZPRD	/	16.	20.	26.
ZRHQ	/	16.	20.	32.
ZSFN	/	16.	20.	32.

COMPLETED

6. LISTING OF THE NATA CODE

The present section consists of a complete source program listing of the NATA code. The routines appear in the same order in which their names are listed at the beginning of Section 5, i.e., main program, block data routines, and then all subroutines and functions in alpha-numeric order of their names.

MAIN PROGRAM FOR NOZZLE FLOW SOLUTION
LOGICAL SUPGO,DATEP,NOTRAN

-375-

COMMON /OUTPUT/ FVOUT(35).GJMF(20)
COMMON /RDOU/ FLOW.FACNAM,CHANAM,LIMOU
COMMON /SS/ CAS.US
COMMON /SPEC/ SPRP(43,30)
COMMON /LN/ ISATOM,ISMOL,JMOL
COMMON /EELEM/ EEP(2,10)
COMMON /THRT/ RSA
DATA NCASE,NERROR,NRCTOT,IRCO /4*0/, ONE /1./, PIV /3.14159/

ISW1A=1
ISW2A=1
ISW3A=1
ISW4A=0
ISW5A=0
ISW6A=0
ISW1B=0
ISW2B=0
IS3=1
ISW4B=0
ISW5B=0
ISW6B=1
IRUN=0
MFILTER=1
NOTRAN=.FALSE.
CALL ELTIME (ET,0)
ISW3D=IS3
CALL READ
IS3=ISW3B
IF (.NOT.NOTRAN) GO TO 20
ISW3B=0
VISC=0.
PR=0.
SIGMA=0.
FLEWIS=0.
CALL LIST
IF (.NOTRAN) WRITE (6,330)
ET=ET
CALL ELTIME (ET,1)
NRCOUT=0
NCASE=NCASE+1
ERR=.FALSE.
OMDST(1)=1.
OMDST(2)=1.
IF (ISW2B.GT.0) SM=0.
CALL INGA
IF (ERR) GO TO 290
IF (ISW2B.LE.0) CALL RESTMP
IF (ERR) GO TO 290
IF (ISW2B.LT.0) GO TO 30
CALL INIT
CALL INTA
IF (ERR) GO TO 290
IF (ISW6A.GE.0) GO TO 40
CALL STUNTS
IRCO=1
GO TO 300

10

20

30

MAI 56
MAI 57
MAI 58
MAI 59
MAI 60
MAI 61
MAI 62
MAI 63
MAI 64
MAI 65
MAI 66
MAI 67
MAI 68
MAI 69
MAI 70
MAI 71
MAI 72
MAI 73
MAI 74
MAI 75
MAI 76
MAI 77
MAI 78
MAI 79
MAI 80
MAI 81
MAI 82
MAI 83
MAI 84
MAI 85
MAI 86
MAI 87
MAI 88
MAI 89
MAI 90
MAI 91
MAI 92
MAI 93
MAI 94
MAI 95
MAI 96
MAI 97
MAI 98
MAI 99
MAI 100
MAI 101
MAI 102
MAI 103
MAI 104
MAI 105
MAI 106
MAI 107
MAI 108
MAI 109
MAI 110

```

40 IF (ISW6A.EQ.2) GO TO 140
   IF (SUPGO.AND.IRCO.EQ.0) GO TO 50
   IF (NOTRAN) GO TO 60
   CALL TRANSX
   IF (ERR) GO TO 290
   IRCO=0
   IF (ISW1B.EQ.0) GO TO 60
   CALL STUNT2
   IF (ISW1B.LT.0) GO TO 140
   CT=TWALL/CTAP
   CALL THERM1
   CPWALL=0.
   DO 70 K=1,NCS
   I=IJS(K)
   CPWALL=CPWALL+QPJ(K)*CCPJ(I)
   CPWALL=CPWALL*CPWALL/CGMW
   WRITE (6,360) CPWALL,TWALL
   IF (NCASE.EQ.1) CALL OUT
   IF (NPRFLS.EQ.2) GO TO 80
   RO=RTHCM(1)
   GO TO 90
80 RO=SQRT(1.27324*RTHCM(1)*RTHCM(2))
90 IF (ISW3P.EQ.0.OR.ISW2B.GT.0.OR.MFITER.EQ.0) GO TO 140
   DO 100 I=1,ISS
   SAVEC(I)=GJA(I)*CMA
   CALL TRANSP (CTAP,PRESA)
   IF (ERR) GO TO 290
   VISROT=VISC/CTAP
   HOPE=CHA*CRP/CMA
   SW=CPWALL*TWALL/HOP-1.
   US=SQRT(8.31436E7*CTAP/CMA)
   CALL EQUIL (1)
   IF (NPRFLS.EQ.1) GO TO 110
   ATH=4.*RTHCM(1)-RO*DELBL(1))*(RTHCM(2)-RO*DELBL(2))
   GO TO 130
110 IF (JDIM.EQ.0) GO TO 120
   ATH=PIV*(RO*OMDST(1))*2
   GO TO 130
120 ATH=2.*RO*OMDST(1)
130 SMASS=453.59*FLOW/ATH
   CALL RESTMP
   IF (ISW2B.LT.0) GO TO 140
   CALL INIT
   CALL INTA
   CF(1)=RHAP*(30.48)**3/453.59
   US=SQRT(8.31436E7*CTAP/CMA)
   CF(2)=US/30.48
   CF(3)=30.48/453.59
   CF(4)=1./2.54
   CF(5)=1.8*CTAP*1.98647/CMA
   CF(6)=RO/2.54
   DO 150 I=1,ISS
   SAVEC(I)=GJA(I)*CMA
   CT=1.
   PRES=1.
150

```

```

MAI 111
MAI 112
MAI 113
MAI 114
MAI 115
MAI 116
MAI 117
MAI 118
MAI 119
MAI 120
MAI 121
MAI 122
MAI 123
MAI 124
MAI 125
MAI 126
MAI 127
MAI 128
MAI 129
MAI 130
MAI 131
MAI 132
MAI 133
MAI 134
MAI 135
MAI 136
MAI 137
MAI 138
MAI 139
MAI 140
MAI 141
MAI 142
MAI 143
MAI 144
MAI 145
MAI 146
MAI 147
MAI 148
MAI 149
MAI 150
MAI 151
MAI 152
MAI 153
MAI 154
MAI 155
MAI 156
MAI 157
MAI 158
MAI 159
MAI 160
MAI 161
MAI 162
MAI 163
MAI 164
MAI 165

```

```

RHO=1.
SU=0.
CH=CHA
CM=CMA
WRITE (6,370)
CALL OUT1
IF (ERR) GO TO 290
CALL ELTIME (ET,1)
IF (ISW6A.EQ.2.OR.ISW18.LT.0) GO TO 300
IF (NOTRAN) GO TO 160
CALL TRANSP (CTAP,PRESA)
IF (ERR) GO TO 290
VISC=VISC*CF(3)
WRITE (6,380) VISC,PR,SIGMA,FLEWIS
IF (ISW6A.NE.0) GO TO 300
VISC=VISC
PR=PR
SIGR=SIGMA
FLEWR=FLEWIS
VISROT=VISC/CTAP
HOP=CHA*CRP/CMA
SW=CPWALL*TWALL/HOP-1.
IF (ISW1A.EQ.IZERO) GO TO 190
IF (.NOT.DATAP) GO TO 180
ISOLN=1
ITYPER=1
XXX(1)=NOZZLE+0.1
XXX(2)=FLOW
XXX(3)=PRESA
XXX(4)=CF(S)*CHA
XXX(5)=DIAM(1)
XXX(6)=CTAP
XXX(7)=CF(1)
XXX(8)=SEN
XXX(9)=CMA
XXX(10)=DIAM(2)
NRCOUT=NRCOUT+1
WRITE (ITPOUT) ITYPER,ISOLN,(XXX(1),I=1,10)
GO TO (180,210,280), ISOLN
CALL FRUZEN
IF (ERR) GO TO 290
CT=CTMAX
PRES=PRESTH
RHO=RHTH
SU=SM/RHTH
CH=CHA-0.5*SU**2
WRITE (6,390)
CALL OUT1
CALL ELTIME (ET,1)
CALL NRMAX
IF (ERR) GO TO 290
PRESTH=PRES
UTH=SU
CHTH=CH
CMTH=CM

```

```

MAI 166
MAI 167
MAI 168
MAI 169
MAI 170
MAI 171
MAI 172
MAI 173
MAI 174
MAI 175
MAI 176
MAI 177
MAI 178
MAI 179
MAI 180
MAI 181
MAI 182
MAI 183
MAI 184
MAI 185
MAI 186
MAI 187
MAI 188
MAI 189
MAI 190
MAI 191
MAI 192
MAI 193
MAI 194
MAI 195
MAI 196
MAI 197
MAI 198
MAI 199
MAI 200
MAI 201
MAI 202
MAI 203
MAI 204
MAI 205
MAI 206
MAI 207
MAI 208
MAI 209
MAI 210
MAI 211
MAI 212
MAI 213
MAI 214
MAI 215
MAI 216
MAI 217
MAI 218
MAI 219
MAI 220

```

```

200 DO 200 I=1,ISS
    GMF(I)=GJ(I)*CM
    IF (ISW3A.EQ.0) GO TO 220
    ISOLN=2
    IF (DATAPE) GO TO 170
    CALL EQUIL (2)
    IF (ERR) GO TO 290
    IM=1
    JJK=0
    ISMCNR=ISMC
    ISSNR=ISS
    WRITE (6,400)
    CT=CTMAX
    PRES=PRESTH
    RHO=RHTH
    SU=UTH
    CH=CHTH
    CM=CMTH
    DO 230 I=1,ISS
        SAVEC(I)=GMF(I)
        CALL OUT1
        IF (ERR) GO TO 290
        CALL ELTIME (ET,1)
        IF (ISW2A.EQ.IZERO) GO TO 300
    DO 240 I=1,ISS
        GJ(I)=GJA(I)
        PRES=1.0
        N=0
        SUM1=ALOG(RHTH)
        A=4.+8.*SUM1
        N=N+1
        IF (N-50) 260,270,270
        SUM2=A+2.
        SUM3=RHTH**A
        D=SUM3*SUM2-2.
        D1=SUM3*(SUM2*SUM1+1.)
        ATEST=A
        A=A-D/D1
        IF (ABS(D).GT..00001) GO TO 250
        IF (ABS(ATEST-A).GT..00001) GO TO 250
        C=(RHTH**2)*(1.-RHTH**A)
        DELT=DELT1
        IF (ISW3B.NE.0) DELT=0.049*(1.-CTMAX)
        CT=1.0-DELT
        CM=CMA
        CALL NEWRAP
        IF (ERR) GO TO 290
        AFNTS=SM/FLUX
        CALL FINDX (AFNTS,-ONE,CX)
        WRITE (6,340) A,C
        ISOLN=3
        IF (DATAPE) GO TO 170
        CALL NONEQ
        CALL ELTIME (ET,1)
        IF (.NOT.ERR) GO TO 300

```

```

MAI 221
MAI 222
MAI 223
MAI 224
MAI 225
MAI 226
MAI 227
MAI 228
MAI 229
MAI 230
MAI 231
MAI 232
MAI 233
MAI 234
MAI 235
MAI 236
MAI 237
MAI 238
MAI 239
MAI 240
MAI 241
MAI 242
MAI 243
MAI 244
MAI 245
MAI 246
MAI 247
MAI 248
MAI 249
MAI 250
MAI 251
MAI 252
MAI 253
MAI 254
MAI 255
MAI 256
MAI 257
MAI 258
MAI 259
MAI 260
MAI 261
MAI 262
MAI 263
MAI 264
MAI 265
MAI 266
MAI 267
MAI 268
MAI 269
MAI 270
MAI 271
MAI 272
MAI 273
MAI 274
MAI 275

```

```

29 CALL DUMPEX
300 NERROR=NERROR+1
    NRCTOT=NRCTOT+NRCTOT
    WRITE (6,420) IRUN,NCASE,NRCOUT,ITPUT,NRCTOT
    ELTMIN=(ET-ET0)/60.
    WRITE (6,410) ELTMIN
    IF (ISW4A) 10,310,10
310 NCOMPL=NCASE-NERROR
    WRITE (6,350) NCOMPL,NERROR
    IF (.NOT.DATAP) GO TO 320
    END FILE ITPUT
    REWIND ITPUT
    CALL EXIT
320 C
    C
    C
330 FORMAT (43H0TRANSPORT PROPERTY CALCULATIONS SUPPRESSED)
340 FORMAT (20H0 DENSITY FIT-ALPHA=.1PE15.7,10H CONSTANT=.E15.7)
350 FORMAT (//15.16H CASES COMPLETED/15.13H CASES FAILED)
360 FORMAT (27H0SPECIFIC HEAT OF COLD GAS=.F7.4,16H BTU/LB-DEG R AT.F8MAI 295
    1.2,6H DEG K)
370 FORMAT (1H1.9X,24H- RESERVOIR CONDITIONS -)
380 FORMAT (//31H RESERVOIR TRANSPORT PROPERTIES//19H VISCOSITY
    1=.3X,1PE9.2,11H LBM/FT-SEC/19H PRANDTL NUMBER =.OPF12.5/19H SIMAI 299
    2GMA =.3X,1PE9.2,7H MMH/CM/19H LEWIS NUMBER =.OPF10. MAI 300
    33)
390 FORMAT (1H1.7X,28H- FROZEN THROAT CONDITIONS -)
400 FORMAT (1H1.6X,33H- EQUILIBRIUM THROAT CONDITIONS -)
410 FORMAT (24H CASE EXECUTION TIME WAS.F6.2,8H MINUTES)
420 FORMAT (8HORUN NO.,17.5X,4HCASE,14.23H OF THIS JOB COMPLETED.,15.2MAI 305
    14H RECORDS WRITTEN ON TAPE.13.1H.,17.36H RECORDS WRITTEN SO FAR INMAI 306
    2 THIS JOB.) MAI 307
    END MAI 308-

```

BLOCK DATA

```

LOGICAL DATAPE, WEDGEM, AXISYM, READXS, REI, AAMS, AXIMOD
DOUBLE PRECISION AIN
COMMON /ELEM/ EP1(2), EP2(2), EP3(2), EP4(2), EP5(2), EP6(2), EP7(2),
1 EP8(2), EP9(2), EP10(2)
COMMON /SPEC/ SP1(43), SP2(43), SP3(43), SP4(43), SP5(43), SP6(43),
1 SP7(43), SP8(43), SP9(43), SP10(43), SP11(43), SP12(43), SP13(43),
2 SP14(43), SP15(43), SP16(43), SP17(43), SP18(43), SP19(43),
3 SP20(43), SP21(43), SP22(43), SP23(43), SP24(43), SP25(43),
4 SP26(43), SP27(43), SP28(43), SP29(43), SP30(43)
COMMON /REAC/ RP1(29), RP2(29), RP3(29), RP4(29), RP5(29), RP6(29),
1 RP7(29), RP8(29), RP9(29), RP10(29), RP11(29), RP12(29), RP13(29),
2 RP14(29), RP15(29), RP16(29), RP17(29), RP18(29), RP19(29),
3 RP20(29), RP21(29), RP22(29), RP23(29), RP24(29), RP25(29),
4 RP26(29), RP27(29), RP28(29), RP29(29), RP30(29), RP31(29),
5 RP32(29), RP33(29), RP34(29), RP35(29), RP36(29), RP37(29),
6 RP38(29), RP39(29), RP40(29), RP41(29), RP42(29), RP43(29),
7 RP44(29), RP45(29), RP46(29), RP47(29), RP48(29), RP49(29),
8 RP50(29), RP51(29), RP52(29), RP53(29), RP54(29), RP55(29),
9 RP56(29), RP57(29), RP58(29), RP59(29), RP60(29), RP61(29),
A RP62(29), RP63(29), RP64(29), RP65(29), RP66(29), RP67(29),
B RP68(29), RP69(29), RP70(29), RP71(29), RP72(29), RP73(29),
C RP74(29), RP75(29), RP76(29), RP77(29), RP78(29), RP79(29),
D RP80(29), RP81(29), RP82(29), RP83(29), RP84(29), RP85(29),
E RP86(29), RP87(29), RP88(29), RP89(29), RP90(29), RP91(29),
F RP92(29)
COMMON /MIXT/ GP1(124), GP2(124), GP3(124), GP4(124), GP5(124),
1 GP6(124)
COMMON /NOZZ/ ZP1(64), ZP2(64), ZP3(64), ZP4(64), ZP5(64), ZP6(64),
1 ZP7(64), ZP8(64), ZP9(64), ZP10(64), ZP11(64), ZP12(64), ZP13(64),
2 ZP14(64), ZP15(64), ZP16(64), ZP17(64), ZP18(64), ZP19(64),
3 ZP20(64)
COMMON /AREA/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12,2), NPARC(2), NPRFLS, NBL
COMMON /MASSFL/ SMASS, CTMXXI, TSTUPI, IS(20)
COMMON /BL/ DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SW, RO, JOIM, IPCINT
COMMON /MODPAR/ XMP1, DXMP, FSTAG, CATFAC, TMODEL, XMODP1, DXMODP,
1 TPLATE, KDIM
COMMON /RDLIST/ IGAS, IGASO, NOZZO, ICHANO
COMMON /MODPT/ TSDIAM(20), TSAR(20), NTS, MBL
COMMON /TAPOUT/ XXX(10), ITPOUT, NRCOUT, IFLOW, ITPYER, IMP, DATAPE
COMMON /BLOUT/ REPF, THE TA(2), SN(2), XSN(2), PRREF, HR, QWDOT(2),
1 TAUW(2)
COMMON /NEWMP/ FACMP, NMODPT
COMMON /RDWEDG/ ANGLE(10), RADLE(5), WX1, DWX, WXI(20), TWEDGE, WK,
1 NWX, NANGLE, NRADLE, WEDGEM, AXISYM, ISW9B
COMMON /CHAN/ CPI(5), CP2(5), CP3(5), CP4(5), CP5(5)
COMMON /AVG/ WSAVE
COMMON /TNE/ TN1(186), TN2(186)
COMMON /RDMOD/ LEWIS, IAMBIP
COMMON /RDMOD/ PRESAI, DELT11, TPRNTI, DELTXI, CXMAXI,
1 CTAPI, XZEROT, PARAMI(3,12,2), BZEROT,
2 HSTAG, READXS, READG, AAMS, AXIMOD,
3 ICASE, NEELS, ISW7B, INT, ICHAN, NOSI, INEQVI, NRECO, JCS(10),

```

```

4  ISCI,ISSI,ISRI,ICI,IE(10),IR(6),ICTS(2,2),IEEP(10)
COMMON /INGNE/ AIN(10,10)
COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DCHLL,DCHRRAT,CCHI,TTEST,
GTEST
1  DATA AIN /100*0./, GAMIN /1.E-10/
DATA CTMXXI /5000./, DELT11,TPRNT1,DELTXI /3*0.01/,CXMAXI /1.E5/,
2  BZEROI /0./, READXS,READG /2*FALSE./, AAMS,AXIMOD /2*TRUE./,
3  ICASE,NEELS,INT,ICHAN,INEQVI,NRECO,ISW7B /7*0/,
4  NQSI /4/
DATA GAMIN /1.E-10/, HTEST /0.01/, TTEST /0.05/, GTEST /0.1/,
1  TETEST /0.05/, QTEST /0.1/, DCHLL /1.E-4/, DCHRRAT /1.E-4/,
2  CCHI /0.1/
DATA WSAVE /3.0/, LEWIS /1/,
DATA ANGLE /10*0./, RADLE /5*0./, NANGLE /0./,
1  NRADLE /0./, WX1,DWX /2*1./, WXI /20*1.E30/, NWX /0./,
2  TWEDGE /30C./, WK /1.333/, ISW9B /0/
DATA NMDDPT /20/, IGAS /1/
DATA YTHETA,SN,XSN,PRREF,HR,REFP,QWDDOT,TAUW /13*0./
DATA ITPUT,DATEPE /8*FALSE./
DATA TSDIAM /20*1.E20/
DATA FSTAG /1./, CATFAC /1./, TMODEL /300./,
1  XMDDPI /1.E20/
DATA TWALL,TPATE /2*30C./, IGASO,NOZZO,ICHAND /3*-11111/
DATA KOIM /1/
DATA JDIM /1/
DATA CTMXXI /5000./, TSTOPI /0./
DATA NPROFL /2*0/, NPRFLS /1/
DATA EP1 /2HE-.5.49597E-4/
DATA EP3 /2HE-.4.0026/
DATA EP4 /1HC.12.011/
DATA EP5 /1HN.14.007/
DATA EP6 /1HO.16./
DATA EP7 /2HAR.39.948/
DATA SP1 /2HE-.1.1.1.1.2*0.1.2*0.6*0.0.1.-14.9276.0.1.1.0.,
1  2.9*0.10*0.3*0./
DATA SP2 /1HN.1.1.5.1.0.0.1.0.0.6*0.112520.1.2944,
1  0.5.1.0.4.10.6.12.18.5*0.0.54962.82456.238270.,
2  250140.5*0.3*0./
DATA SP3 /1HO.1.1.6.1.0.0.1.0.0.6*0.58990.1.4938.0.,
1  7.1.0.5.3.1.5.1.8.24.3*0.0.453.648.45367.,
2  96616.21399.249820.3*0.3*0./
DATA SP4 /2HAR.1.1.7.1.0.0.1.0.0.6*0.0.1.1.8663.0.1.1.0.,
1  1.9*0.10*0.3*0./
DATA SP5 /2HN2.1.1.5.1.0.0.2.0.0.3.451483.3.088332E-4,
1  -4.251428E-8.2.739295E-12.-5.468320E-17.3.071269.0.2.,
2  -4.106.3352.5.1.1.1.3.6.1.2.5*0.0.143540.170480.,
3  171500.198110.5*0.3*0./
DATA SP6 /2HO2.1.1.6.1.0.0.2.0.0.3.249473.4.963449E-4,
1  -6.701753E-8.4.443339E-12.-1.000281E-16.5.915022.0.2.,
2  .1140.2239.5.1.1.1.3.2.1.3.3.5*0.0.22639.37726.,
3  103200.142390.5*0.3*0./
DATA SP7 /2HNO.2.1.5.1.0.0.1.0.0.3.756216.2.083961E-4,
1  -2.639548E-8.1.690332E-12.-3.611523E-17.3.611167.21460.2.,
2  .5455.2699.7.1.1.1.2.2.2.4.2.2.3*0.0.346.,
3  125700.131320.149100.151770.173340.3*0.3*0./

```

E-
HE
CNO AR

E-
EN
N
N
N
N
O
O
O
O
AR
AR
N2
N2
N2
N2
O2
O2
O2
NO
NO
NO

1	2*0..2*1..C..6..1.5..1.6..1.4..1.13..1.14..1.15..1.4*0./	REAC	008
1	DATA RP9 /7.8E14,-1.5.150C50..1.1.1.2.1.7.1.2*0..2.1.3.1.0..1..	REAC	009
1	2*0..2*1..0..4..1.3..1.2.1.7.1.17.1.6*0./	REAC	009
1	DATA RP10 /3.2E13,1..39150..0..2.1.2.1.3.1.7.1.0..2.1.6.1.0..1..	REAC	010
1	1.0..1.0..0..0..0..10*0./	REAC	010
1	DATA RP11 /7.E13.C.75510..0..2.1.2.1.3.1.5.1.0..2.1.7.1.0..1..	REAC	011
1	1.0..1.0..0..0..0..10*0./	REAC	011
1	DATA RP12 /6.7E15,-1.5.0..0..2.1.2.1.8.1.1.1.0..2.1.3.1.0..1..	REAC	012
1	0..1.1.0..0..0..10*0./	REAC	012
1	DATA RP13 /2.2E22,-4.5.C..C..2.1.2.1.10.1.1.1.0..3.1.1.1.0..1..	REAC	013
1	2.0..1.1.0..0..0..10*0./	REAC	013
1	DATA RP14 /2.2E22,-4.5.0..0..2.1.2.1.9.1.1.1.0..2.1.1.1.0..1..	REAC	014
1	2.0..1.1.0..0..0..10*0./	REAC	014
1	DATA RP15 /8.E15,-1.5.C..0..2.1.1.1.12.1.1.1.0..3.1.2*0..1.1.0..	REAC	015
1	2.2*0..C..10*0./	REAC	015
1	DATA RP16 /7.8E13.C.5.0..0..2.1.2.1.6.1.10.1.0..3.1.12.1.0..1..	REAC	016
1	1.0..1.1.0..0..0..10*0./	REAC	016
1	DATA RP17 /7.8E13.C.5.C..0..2.1.2.1.2.1.11.1.0..5.1.9.1.0..1..	REAC	017
1	0..1.1.0..0..0..10*0./	REAC	017
1	DATA RP18 /1.5E16,-1.5.0..0..2.1.1.1.1.1.1.1.0..2.1.2*0..1.1..	REAC	018
1	0..2.2*0..C..10*0./	REAC	018
1	DATA RP19 /1.E14,-2.5.C..0..3.1.2.1.7.1.8.1.1.1.5.1.6.1.0..1..	REAC	019
1	1.1.1.0..0..0..10*0./	REAC	019
1	DATA RP20 /2.2E16,-2.5.0..0..3.1.2.1.8.1.1.1.5.1.7.1.5.1.C..1..	REAC	020
1	1.1.1.1.0..0..0..10*0./	REAC	020
1	DATA RP21 /1.5E13.C..0..0..2.1.2.1.7.1.10.1.0..3.1.8.1.0..1..	REAC	021
1	0..1.1.0..0..0..10*0./	REAC	021
1	DATA RP22 /3.4E11,-2..45700..0..2.1.2.1.5.1.10.1.0..3.1.11.1.0..	REAC	022
1	1.1.C..1.1.0..0..0..10*0./	REAC	022
1	DATA RP23 /4.8E14.C..0..0..2.1.2.1..1.9.1.0..2.1.8.1.0..1.1.0..	REAC	023
1	1.1.0..0..0..10*0./	REAC	023
1	DATA RP24 /1.8E15.C.5.C..0..2.1.2.1.7.1.12.1.0..6.1.8.1.0..1.1..	REAC	024
1	0..1.1.0..0..0..10*0./	REAC	024
1	DATA RP25 /1.E14.0..0..0..2.1.2.1.6.1.9.1.0..3.1.8.1.0..1.1.0..	REAC	025
1	1.1.0..0..0..10*0./	REAC	025
1	DATA RP26 /8.E16,-2.5.C..0..3.1.2.1.8.1.1.1.6.1.7.1.6.1.0..1..	REAC	026
1	1.1.1.0..0..0..10*0./	REAC	026
1	DATA RP35 /5.46E21,-4.3.0..0..2.1.1.1.1.1.19.1.0..21.1.2*0..2*1..	REAC	035
1	0..1.2*0..1.1.1.9*0./	REAC	035
1	DATA RP36 /1.82E21,-4.3.0..0..2.1.1.1.1.1.19.1.0..22.1.2*0..2*1..	REAC	036
1	0..1.2*0..1.1.1.9*0./	REAC	036
1	DATA RP37 /1.27E11,-0.81.0..0..2.1.1.1.1.1.19.1.0..21.1.2*0..1..	REAC	037
1	1.0..1.2*0..0..10*0./	REAC	037
1	DATA RP38 /3.80E10,-0.85.0..0..2.1.1.1.1.1.19.1.0..22.1.2*0..2*1..	REAC	038
1	0..1.2*0..0..10*0./	REAC	038
1	DATA RP39 /3.92E16,2*0..1..2.1.1.1.16.1.19.1.0..23.1.2*0..2*1..	REAC	039
1	0..1.2*0..1.1.16.1.9*0./	REAC	039
1	DATA RP40 /1.54E21,-4.3.0..0..2.1.1.1.1.1.23.1.0..24.1.2*0..1.1..	REAC	040
1	0..1.2*0..1.1.1.9*0./	REAC	040
1	DATA RP41 /5.13E20,-4.3.0..0..2.1.1.1.1.1.23.1.0..16.1.2*0..2*1..	REAC	041
1	0..2.2*0..1.1.1.9*0./	REAC	041
1	DATA RP42 /2.26E14,3*0..2*2.1.1.1.23.1.0..16.1.21.1.0..2*1.0..	REAC	042
1	1.1.0..0..0..10*0./	REAC	042
1	DATA RP43 /7.5E13,3*0..2*2.1.1.1.23.1.0..16.1.22.1.0..2*1.0..1..	REAC	043
1	1.0..0..0..10*0./	REAC	043

DATA	RP44	/3.65E16,-0.5*0.,1.2*1.1,22.1,2*0.,21.1,2*0.,1.2*0.,1.2*0.,	REAC	044	
1	1.2*0.,1.1,1.1,9*0.,		REAC	044	
1	DATA	RP45	/8.E14,-0.25,1272.,1.2*1.1,21.1,2*0.,16.1,2*0.,1.2*0.,	REAC	045
1	1.2*0.,1.1,1.1,9*0.,		REAC	045	
1	DATA	RP46	/8.E14,-0.25,1272.,1.2*1.1,22.1,2*0.,16.1,2*0.,1.2*0.,	REAC	046
1	1.2*0.,1.1,1.1,9*0.,		REAC	046	
1	DATA	RP47	/5.20E10,0.,1590.,1.2*1.1,22.1,2*0.,16.1,2*0.,1.2*0.,	REAC	047
1	1.2*0.,1.1,1.1,9*0.,		REAC	047	
1	DATA	RP48	/1.87E15,0.,167.0.,0.,1.1,3.1,21.1,2*0.,1.1,16.1,19.1,	REAC	048
1	2.2*0.,3.1,0.,10*0.,		REAC	048	
1	DATA	RP49	/5.05E15,0.,167.2*0.,2.1,3.1,21.1,22.1,0.,1.1,16.1,	REAC	049
1	19.1,2*1.0,3.1,0.,10*0.,		REAC	049	
1	DATA	RP50	/5.28E15,0.,167.2*0.,1.1,3.1,22.1,2*0.,1.1,16.1,19.1,	REAC	050
1	2.2*0.,3.1,0.,10*0.,		REAC	050	
1	DATA	RP51	/5.2E14,0.,5.0.,1.2,1.1,1.1,16.1,21.1,0.,24.1,2*0.,2*1.,	REAC	051
1	0.,1.2*0.,1.1,1.1,16.1,9*0.,		REAC	051	
1	DATA	RP52	/1.87E15,0.,167.2*0.,1.1,3.1,24.1,2*0.,1.1,16.1,19.1,	REAC	052
1	2.2*0.,1.1,3.1,0.,10*0.,		REAC	052	
1	DATA	RP53	/8.E14,-0.25,1272.,1.2*1.1,24.1,2*0.,16.1,2*0.,1.2*0.,	REAC	053
1	2.2*0.,1.1,1.1,9*0.,		REAC	053	
1	DATA	RP54	/4.48E15,-1.,256000.,1.1,1.2,1.1,14.1,2*0.,17.1,3.1,0.,	REAC	054
1	1.2*0.,2.1,0.,9.1,2.1,3.1,4.1,5.1,6.1,13.1,14.1,15.1,17.1,		REAC	054	
2	C.,		REAC	054	
1	DATA	RP55	/8.81E14,-2.,125600.,1.1,1.2,1.1,13.1,2*0.,3.1,14.1,0.,	REAC	055
1	1.2*0.,2.1,0.,9.1,2.1,3.1,4.1,5.1,6.1,13.1,14.1,15.1,17.1,		REAC	055	
2	0.,		REAC	055	
1	DATA	RP56	/2.33E11,5.130500.,0.,1.1,2.1,14.1,2*0.,13.1,17.1,0.,	REAC	056
1	2.2*0.,2.1,0.,11*0.,		REAC	056	
1	DATA	RP57	/2.73E13,5.138100.,0.,2.1,2.1,3.1,14.1,0.,6.1,17.1,0.,	REAC	057
1	2*1.0,2.2*1.0,11*0.,		REAC	057	
1	DATA	RP58	/2.86E13,5.106500.,0.,2.1,2.1,2.1,14.1,0.,7.1,17.1,0.,	REAC	058
1	2*1.0,2.2*1.0,11*0.,		REAC	058	
1	DATA	RP59	/4.59E10,5.23970.,0.,2.1,2.1,7.1,14.1,0.,13.,2.1,0.,	REAC	059
1	2*1.0,2.2*1.0,11*0.,		REAC	059	
1	DATA	RP60	/2.55E11,5.7606.,0.,2.1,2.1,13.1,3.1,0.,6.1,14.1,0.,	REAC	060
1	2*1.0,2.2*1.0,11*0.,		REAC	060	
1	DATA	RP61	/1.07E14,5.67050.,0.,2.1,2.1,14.1,25.1,0.,13.1,18.1,0.,	REAC	061
1	2*1.0,2.2*1.0,11*0.,		REAC	061	
1	DATA	RP62	/6.03E13,5.63360.,0.,2.1,2.1,14.1,18.1,0.,17.1,25.1,0.,	REAC	062
1	2*1.0,2.2*1.0,11*0.,		REAC	062	
1	DATA	RP63	/6.66E14,5.54160.,0.,2.1,2.1,3.1,18.1,0.,17.1,10.1,0.,	REAC	063
1	2*1.0,2.2*1.0,11*0.,		REAC	063	
1	DATA	RP64	/1.09E14,5.9222.,0.,2.1,2.1,14.1,10.1,0.,3.1,25.1,0.,	REAC	064
1	2*1.0,2.2*1.0,11*0.,		REAC	064	
1	DATA	RP65	/5.47E14,5.74700.,0.,2.1,2.1,3.1,25.1,0.,6.1,18.1,0.,	REAC	065
1	2*1.0,2.2*1.0,11*0.,		REAC	065	
1	DATA	RP66	/1.5E16,-1.5,0.,0.,2.1,2.1,25.1,1.1,0.,17.1,3.1,0.,	REAC	065
1	2*1.0,2.2*1.0,11*0.,		REAC	065	
1	DATA	RP67	/2.2E22,-4.5,0.,0.,2.1,2.1,18.1,1.1,0.,17.1,1.1,0.,	REAC	067
1	1.2,0.,2.2*1.0,11*0.,		REAC	067	
1	DATA	RP68	/4.59E14,5.10900.,0.,2.1,2.1,14.1,8.1,0.,25.1,7.1,0.,	REAC	068
1	2*1.0,2.2*1.0,11*0.,		REAC	068	
1	DATA	RP69	/4.53E14,5.44490.,0.,2.1,2.1,14.1,12.1,0.,25.1,6.1,0.,	REAC	069
1	2*1.0,2.2*1.0,11*0.,		REAC	069	
1	DATA	RP70	/5.96E14,5.3227.,0.,2.1,2.1,17.1,8.1,0.,25.1,2.1,0.,	REAC	070

6	2.857325.	-0.052408.	0.0	0.0	2.984499.	0.0	0.127000.	0.0	0.127000.	NOZZ	3
7	2.984499.	0.0	0.0	0.0	2.853019.	0.0	0.0	0.0	0.0	NOZZ	3
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	3
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	3
A	4HMRA										
1	MRA 1.00-INCH	NOZZLE									
2	DATA ZP 4 /	1.27000.	3.1.	1.1.	2.1.	6.1.	2.1.	1.1.	0.1.	0.1.	0.1.
3	-3.528553.	-3.308723.	0.0	0.0	-0.559095.	0.0	0.0	0.0	0.0	0.0	0.1.
4	0.032870.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.015062.	-0.17327.	0.0	0.0	5.324564.	0.0	0.0	0.0	0.0	0.0	0.317500.
6	-0.247563.	-1.732051.	0.0	0.0	1.933256.	0.0	0.0	0.0	0.0	0.0	0.635000.
7	1.269425.	-0.052408.	0.0	0.0	1.397070.	0.0	0.0	0.0	0.0	0.0	0.127000.
8	1.297000.	0.0	0.0	0.0	1.265519.	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A	4HMRA										
1	ED: 0.32-INCH	NOZZLE									
2	DATA ZP 5 /	0.40640.	3.1.	1.1.	2.1.	6.1.	4.1.	1.1.	3.1.	1.1.	0.1.
3	-4.836277.	-4.806674.	0.0	0.0	-1.720592.	0.0	0.0	0.0	0.0	0.0	0.0
4	0.058642.	1.134806.	0.0	0.0	1.141119.	0.0	0.0	0.0	0.0	0.0	0.127000.
5	1.725971.	-0.176327.	0.0	0.0	0.0	2.453567.	0.0	0.0	0.0	0.0	0.127000.
6	0.429518.	-0.445229.	0.0	0.0	0.0	1.079555.	0.0	0.0	0.0	0.0	0.127000.
7	0.393271.	-0.466308.	0.0	0.0	0.0	0.533400.	0.0	0.0	0.0	0.0	0.127000.
8	0.533400.	0.0	0.0	0.0	0.127000.	0.0	0.0	0.0	0.0	0.0	0.0
9	0.869315.	1.193448.	0.0	0.0	0.127000.	0.0	0.0	0.0	0.0	0.0	0.0
A	4HEOS										
1	ED: 0.775-INCH	NOZZLE									
2	DATA ZP 6 /	0.98171.	3.1.	1.1.	2.1.	8.1.	2.1.	1.1.	2.1.	1.1.	0.1.
3	-5.974197.	-5.544594.	0.0	0.0	-2.863580.	0.0	0.0	0.0	0.0	0.0	0.1.
4	0.006947.	0.0	0.0	0.0	0.052330.	0.0	0.0	0.0	0.0	0.0	-2.735612.
5	1.509393.	-0.176327.	0.0	0.0	0.0	2.437734.	0.0	0.0	0.0	0.0	0.127000.
6	-0.292950.	-0.445229.	0.0	0.0	0.0	1.065978.	0.0	0.0	0.0	0.0	0.127000.
7	0.470487.	-0.577350.	0.0	0.0	0.0	1.251729.	0.0	0.0	0.0	0.0	0.127000.
8	0.981535.	-0.052408.	0.0	0.0	0.0	1.108710.	0.0	0.0	0.0	0.0	0.127000.
9	1.108710.	0.0	0.0	0.0	0.127000.	0.0	0.0	0.0	0.0	0.0	0.0
A	4HEOS										
1	MRA 0.75-INCH	NOZZLE									
2	DATA ZP 7 /	0.95250.	3.1.	1.1.	2.1.	6.1.	2.1.	1.1.	0.1.	0.1.	0.1.
3	-3.624795.	-3.413759.	0.0	0.0	-0.999293.	0.0	0.0	0.0	0.0	0.0	0.0
4	0.032870.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	4.984842.	-0.176327.	0.0	0.0	0.0	5.323822.	0.0	0.0	0.0	0.0	0.304800.
6	-0.436579.	-1.732050.	0.0	0.0	0.0	1.511746.	0.0	0.0	0.0	0.0	0.635000.
7	0.952325.	-0.052408.	0.0	0.0	0.0	1.079499.	0.0	0.0	0.0	0.0	0.127000.
8	1.079499.	0.0	0.0	0.0	0.127000.	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.949019.	0.0	0.0	0.0	0.0	0.0
A	4HMRA										
1	MRA 1.5-INCH	NOZZLE									
2	DATA ZP 8 /	1.90500.	3.1.	1.1.	2.1.	6.1.	2.1.	1.1.	2.1.	1.1.	0.1.
3	-3.17500.	-3.17500.	0.0	0.0	-3.17500.	0.0	0.0	0.0	0.0	0.0	0.0

1	1.1.	3.1.	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	0.1.	0.1.	0.1.	0.1.	0.1.	0.1.	NOZZ	8
2	-3.157434.	-2.946399.	0.0	-1.075492.	0.0	-0.558800.	-0.006647.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
3	0.032870.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
4	5.082271.	-0.176327.	0.0	0.0	5.338842.	-3.210362.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
5	0.387933.	-1.732050.	0.0	0.0	2.568238.	-0.525567.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
6	1.904823.	-0.052408.	0.0	0.0	2.031999.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
7	2.0031999.	0.0	0.0	0.127000.	1.900517.	0.267949.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	8
A	4H MRA															NOZZ	8
1	10MW 2.25-INCH NOZZLE															NOZZ	9
2	DATA ZP 9 /	2.85750.														NOZZ	9
3	1.1.	3.1.	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	0.1.	0.1.	0.1.	0.1.	0.1.	0.1.	NOZZ	9
4	-7.918200.	-7.698371.	0.0	-8.00100.	0.0	-5.714999.	-0.013283.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	9
5	0.039444.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	9
6	3.654496.	-0.176327.	0.0	0.0	4.739010.	-7.973334.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	9
7	-3.437203.	-1.732050.	0.0	0.0	5.570952.	-5.521296.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	9
8	2.856915.	-0.087489.	0.0	0.0	3.009897.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	9
9	3.009897.	0.0	0.0	0.152400.	2.852120.	0.267949.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	9
A	4H10MW/															NOZZ	9
1	EOS 1.09-INCH NOZZLE															NOZZ	10
2	DATA ZP10 /	1.38176.														NOZZ	10
3	1.1.	3.1.	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	0.1.	0.1.	0.1.	0.1.	0.1.	0.1.	NOZZ	10
4	-3.540351.	-3.473112.	0.0	-3.55600.	0.0	-2.393882.	-1.280036.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	10
5	0.005397.	0.0	0.0	0.006574.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	10
6	1.921388.	-0.176327.	0.0	0.0	2.295506.	-3.594458.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	10
7	0.824850.	-0.487733.	0.0	0.0	2.243805.	-2.319619.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	10
8	1.269022.	-0.305731.	0.0	0.0	1.903271.	-1.205774.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	10
9	1.381153.	-0.217432.	0.0	0.0	1.407157.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	10
A	4H EOS															NOZZ	10
1	DCA T12 CHANNEL, PROFILE 1															NOZZ	11
2	DATA ZP11 /	2.54000.														NOZZ	11
3	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	0.1.	0.1.	0.1.	0.1.	0.1.	0.1.	NOZZ	11
4	-6.730998.	-6.095999.	0.0	-7.62000.	0.0	-6.095999.	-6.095999.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	11
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	11
6	-1.175995.	-0.577350.	0.0	0.0	3.809999.	-6.095999.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	11
7	2.539999.	0.0	0.0	102.540000.	105.079999.	0.0	102.540000.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	11
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	11
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	11
A	4H DCA															NOZZ	11
1	DCA T12 CHANNEL, PROFILE 2															NOZZ	12
2	DATA ZP12 /	1.27000.														NOZZ	12
3	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	2.1.	1.1.	0.1.	0.1.	0.1.	0.1.	0.1.	0.1.	NOZZ	12
4	-6.771664.	0.0	0.0	-7.62000.	0.0	-6.463396.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	12
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	12
6	-0.501576.	-0.495987.	0.0	0.0	16.509995.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	12
7	16.509995.	0.0	0.0	15.240000.	1.034893.	0.176327.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	12
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	12
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	NOZZ	12

BLOCK DATA		COMMON /TRANS2/		COMMON /TRANV/		COMMON /TRANJ/		COMMON /TRANI/	
1	BD2	1	BD2	1	BD2	1	BD2	1	BD2
2	BD2	2	BD2	2	BD2	2	BD2	2	BD2
3	BD2	3	BD2	3	BD2	3	BD2	3	BD2
4	BD2	4	BD2	4	BD2	4	BD2	4	BD2
5	BD2	5	BD2	5	BD2	5	BD2	5	BD2
6	BD2	6	BD2	6	BD2	6	BD2	6	BD2
7	BD2	7	BD2	7	BD2	7	BD2	7	BD2
8	BD2	8	BD2	8	BD2	8	BD2	8	BD2
9	BD2	9	BD2	9	BD2	9	BD2	9	BD2
10	BD2	10	BD2	10	BD2	10	BD2	10	BD2
11	BD2	11	BD2	11	BD2	11	BD2	11	BD2
12	BD2	12	BD2	12	BD2	12	BD2	12	BD2
13	BD2	13	BD2	13	BD2	13	BD2	13	BD2
14	BD2	14	BD2	14	BD2	14	BD2	14	BD2
15	BD2	15	BD2	15	BD2	15	BD2	15	BD2
16	BD2	16	BD2	16	BD2	16	BD2	16	BD2
17	BD2	17	BD2	17	BD2	17	BD2	17	BD2
18	BD2	18	BD2	18	BD2	18	BD2	18	BD2
19	BD2	19	BD2	19	BD2	19	BD2	19	BD2
20	BD2	20	BD2	20	BD2	20	BD2	20	BD2
21	BD2	21	BD2	21	BD2	21	BD2	21	BD2
22	BD2	22	BD2	22	BD2	22	BD2	22	BD2
23	BD2	23	BD2	23	BD2	23	BD2	23	BD2
24	BD2	24	BD2	24	BD2	24	BD2	24	BD2
25	BD2	25	BD2	25	BD2	25	BD2	25	BD2
26	BD2	26	BD2	26	BD2	26	BD2	26	BD2
27	BD2	27	BD2	27	BD2	27	BD2	27	BD2
28	BD2	28	BD2	28	BD2	28	BD2	28	BD2
29	BD2	29	BD2	29	BD2	29	BD2	29	BD2
30	BD2	30	BD2	30	BD2	30	BD2	30	BD2
31	BD2	31	BD2	31	BD2	31	BD2	31	BD2
32	BD2	32	BD2	32	BD2	32	BD2	32	BD2
33	BD2	33	BD2	33	BD2	33	BD2	33	BD2
34	BD2	34	BD2	34	BD2	34	BD2	34	BD2
35	BD2	35	BD2	35	BD2	35	BD2	35	BD2
36	BD2	36	BD2	36	BD2	36	BD2	36	BD2
37	BD2	37	BD2	37	BD2	37	BD2	37	BD2
38	BD2	38	BD2	38	BD2	38	BD2	38	BD2
39	BD2	39	BD2	39	BD2	39	BD2	39	BD2
40	BD2	40	BD2	40	BD2	40	BD2	40	BD2
41	BD2	41	BD2	41	BD2	41	BD2	41	BD2
42	BD2	42	BD2	42	BD2	42	BD2	42	BD2
43	BD2	43	BD2	43	BD2	43	BD2	43	BD2
44	BD2	44	BD2	44	BD2	44	BD2	44	BD2
45	BD2	45	BD2	45	BD2	45	BD2	45	BD2
46	BD2	46	BD2	46	BD2	46	BD2	46	BD2
47	BD2	47	BD2	47	BD2	47	BD2	47	BD2
48	BD2	48	BD2	48	BD2	48	BD2	48	BD2
49	BD2	49	BD2	49	BD2	49	BD2	49	BD2
50	BD2	50	BD2	50	BD2	50	BD2	50	BD2
51	BD2	51	BD2	51	BD2	51	BD2	51	BD2
52	BD2	52	BD2	52	BD2	52	BD2	52	BD2
53	BD2	53	BD2	53	BD2	53	BD2	53	BD2
54	BD2	54	BD2	54	BD2	54	BD2	54	BD2

56	BD2	II66(S).	II67(S).	II68(S).	II69(S).	II70(S).
57	BD2	II71(S).	II72(S).	II73(S).	II74(S).	II75(S).
58	BD2	II76(S).	II77(S).	II78(S).	II79(S).	II80(S).
59	BD2	II81(S).	II82(S).	II83(S).	II84(S).	II85(S).
60	BD2	II86(S).	II87(S).	II88(S).	II89(S).	II90(S).
61	BD2	II91(S).	II92(S).	II93(S).	II94(S).	II95(S).
62	BD2	II96(S).	II97(S).	II98(S).	II99(S).	II100(S).
63	BD2	COMMON /TF NSA/				
64	BD2	1 TL 1(100).	TL 2(100).	TL 3(100).	TL 4(100).	TL 5(100).
65	BD2	1 TL 7(100).	TL 8(100).	TL 9(100).	TL 10(100).	TL 11(100).
66	BD2	1 TL 13(100).	TL 14(100).	TL 15(100).	TL 16(100).	TL 17(100).
67	BD2	1 TL 19(100).	TL 20(100).	TL 21(100).	TL 22(100).	TL 23(100).
68	BD2	1 TL 25(100).	TL 26(100).	TL 27(100).	TL 28(100).	TL 29(100).
69	BD2	1 TL 31(100).	TL 32(100).	TL 33(100).	TL 34(100).	TL 35(100).
70	BD2	COMMON /TRANS/ NV(1).N				
71	BD2	COMMON /RDTR/ ISW8B				
72	BD2	DATA IS*58 /O/				
73	BD2	DATA NV/3.2.3.4.3.2.0.3.5.0.2.4.3.3.0.				
74	BD2	DATA NNKQ/3/				
75	BD2	DATA ISEQ/1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16.				
76	BD2	1 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32.				
77	BD2	2 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48.				
78	BD2	3 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 37*0/BD2				
79	BD2	DATA KKQ/3. 3. 3. 3. 6. 14. 14. 14. 5. 6. 6. 5. 5. 5. 5. 5.				
80	BD2	1 5. 5. 5. 5. 9. 14. 14. 14. 14. 14. 14. 14. 14. 14. 14. 14.				
81	BD2	2 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5.				
82	BD2	3 37*0/				
83	BD2	DATA NNQ/3. 1. 1. 1. 3. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.				
84	BD2	1 2. 1. 4. 4. 2. 2. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.				
85	BD2	2 1. 2. 2. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.				
86	BD2	3 37*0/				
87	BD2	DATA II 1/ 4.26.27. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
88	BD2	DATA JJ 1/ 4.26.27. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
89	BD2	DATA VV 1/2.553E7. 292. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
90	BD2	DATA II 2/ 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
91	BD2	DATA JJ 2/20. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
92	BD2	DATA VV 2/6.244E10. 1969. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
93	BD2	DATA II 3/ 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
94	BD2	DATA JJ 3/ 5. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
95	BD2	DATA VV 3/1.497E7. 327. 1.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.				
96	BD2	DATA II 4/ 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
97	BD2	DATA JJ 4/ 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
98	BD2	DATA VV 4/ 2.414E6. 764. 1.0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
99	BD2	DATA II 5/17. 2. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
100	BD2	DATA JJ 5/17.17. 17. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
101	BD2	DATA VV 5/999. 6.88. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
102	BD2	DATA II 6/17. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
103	BD2	DATA JJ 6/17. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
104	BD2	DATA VV 6/173. 1.115. 1.0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
105	BD2	DATA II 7/ 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
106	BD2	DATA JJ 7/17. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
107	BD2	DATA VV 7/166. 1.1145. 1.0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
108	BD2	DATA II 8/ 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
109	BD2	DATA JJ 8/17. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				
110	BD2	DATA II 9/17. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.				

DATA VV 8/142.. 1.1408. 1.0. 0.. 0.. 0./
DATA II 9/13. 0. 0. 0. 0./
DATA JJ 9/17. 0. 0. 0. 0./
DATA VV 9/1. 0. 51.. 2.. 0.. 0./
DATA II 10/2. 0. 0. 0. 0./
DATA JJ 10/9. 0. 0. 0. 0./
DATA VV 10/998.. 4.824. 0.. 0.. 0./
DATA II 11/3. 0. 0. 0. 0./
DATA JJ 11/10. 0. 0. 0. 0./
DATA VV 11/997.. 4.722. 0.. 0.. 0./
DATA II 12/2. 0. 0. 0. 0./
DATA JJ 12/2. 0. 0. 0. 0./
DATA VV 12/1. 0. 81.. 20.. 0.. 0./
DATA II 13/2. 2. 2. 0. 0./
DATA JJ 13/15. 7. 5. 0. 0./
DATA VV 13/1. 0. 101.. 20.. 0.. 0./
DATA II 14/3. 0. 0. 0. 0./
DATA JJ 14/1. 0. 121.. 20.. 0.. 0./
DATA VV 15/14. 13. 5. 6. 7./
DATA II 15/14. 14. 14. 14. 14./
DATA VV 15/1. 0. 141.. 20.. 0.. 0./
DATA II 16/14. 13. 7. 13. 5./
DATA VV 16/15. 15. 13. 13. 13./
DATA JJ 16/1. 0. 141.. 20.. 0.. 0./
DATA II 17/6. 7. 5. 5. 5./
DATA JJ 17/13. 7. 7. 15. 5./
DATA VV 17/1. 0. 141.. 20.. 0.. 0./
DATA II 18/3. 3. 3. 2. 0./
DATA JJ 18/5. 15. 7. 6. 0./
DATA VV 18/1. 0. 161.. 20.. 0.. 0./
DATA II 19/6. 6. 5. 5. 7./
DATA JJ 19/15. 7. 15. 7. 7./
DATA VV 19/1. 0. 181.. 20.. 0.. 0./
DATA II 20/7. 5. 0. 0. 0./
DATA JJ 20/7. 6. 0. 0. 0./
DATA VV 20/1. 0. 181.. 20.. 0.. 0./
DATA II 21/3. 0. 0. 0. 0./
DATA JJ 21/3. 0. 0. 0. 0./
DATA VV 21/1. 0. 201.. 20.. 0.. 0./
DATA II 22/3. 3. 2. 3. 0./
DATA JJ 22/15. 7. 6. 6. 0./
DATA VV 22/1. 0. 221.. 20.. 0.. 0./
DATA II 23/6. 6. 6. 7. 0./
DATA JJ 23/15. 7. 6. 7. 0./
DATA VV 23/1. 0. 241.. 20.. 0.. 0./
DATA II 24/2. 2. 0. 0. 0./
DATA JJ 24/15. 7. 0. 0. 0./
DATA VV 24/2. 0. 6. 0. 0.5. 1.0. 1.0./
DATA II 25/15. 7. 0. 0. 0./
DATA JJ 25/0. 5. 1. 0. 0.. 0./
DATA VV 26/2. 2. 0. 0. 0./
DATA II 26/5. 3. 0. 0. 0./
DATA VV 26/1. 0. 1. 0. 0.. 0./

BD2 111
BD2 112
BD2 113
BD2 114
BD2 115
BD2 116
BD2 117
BD2 118
BD2 119
BD2 120
BD2 121
BD2 122
BD2 123
BD2 124
BD2 125
BD2 126
BD2 127
BD2 128
BD2 129
BD2 130
BD2 131
BD2 132
BD2 133
BD2 134
BD2 135
BD2 136
BD2 137
BD2 138
BD2 139
BD2 140
BD2 141
BD2 142
BD2 143
BD2 144
BD2 145
BD2 146
BD2 147
BD2 148
BD2 149
BD2 150
BD2 151
BD2 152
BD2 153
BD2 154
BD2 155
BD2 156
BD2 157
BD2 158
BD2 159
BD2 160
BD2 161
BD2 162
BD2 163
BD2 164
BD2 165

DATA	II	27/14.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	27/14.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	27/14.	0.	0.	0.	0.	0.	0.	0.
DATA	II	28/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	28/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	28/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	29/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	29/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	29/15.	0.	0.	0.	0.	0.	0.	0.
DATA	II	30/14.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	30/14.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	30/14.	0.	0.	0.	0.	0.	0.	0.
DATA	II	31/14.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	31/14.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	31/14.	0.	0.	0.	0.	0.	0.	0.
DATA	II	32/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	32/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	32/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	33/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	33/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	33/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	34/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	34/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	34/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	35/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	35/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	35/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	36/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	36/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	36/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	37/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	37/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	37/15.	0.	0.	0.	0.	0.	0.	0.
DATA	II	38/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	38/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	38/15.	0.	0.	0.	0.	0.	0.	0.
DATA	II	39/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	39/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	39/15.	0.	0.	0.	0.	0.	0.	0.
DATA	II	40/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	40/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	40/15.	0.	0.	0.	0.	0.	0.	0.
DATA	II	41/14.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	41/14.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	41/14.	0.	0.	0.	0.	0.	0.	0.
DATA	II	42/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	42/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	42/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	43/13.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	43/13.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	43/13.	0.	0.	0.	0.	0.	0.	0.
DATA	II	44/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	44/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	44/15.	0.	0.	0.	0.	0.	0.	0.
DATA	II	45/15.	0.	0.	0.	0.	0.	0.	0.
DATA	JJ	45/15.	0.	0.	0.	0.	0.	0.	0.
DATA	VV	45/15.	0.	0.	0.	0.	0.	0.	0.

BD2	166
BD2	167
BD2	168
BD2	169
BD2	170
BD2	171
BD2	172
BD2	173
BD2	174
BD2	175
BD2	176
BD2	177
BD2	178
BD2	179
BD2	180
BD2	181
BD2	182
BD2	183
BD2	184
BD2	185
BD2	186
BD2	187
BD2	188
BD2	189
BD2	190
BD2	191
BD2	192
BD2	193
BD2	194
BD2	195
BD2	196
BD2	197
BD2	198
BD2	199
BD2	200
BD2	201
BD2	202
BD2	203
BD2	204
BD2	205
BD2	206
BD2	207
BD2	208
BD2	209
BD2	210
BD2	211
BD2	212
BD2	213
BD2	214
BD2	215
BD2	216
BD2	217
BD2	218
BD2	219
BD2	220

-396-

42.8000E	04.3.0000E	04.1.0000E	03.2.0000E	03.3.0000E	03.4.0000E	03.802
55.0000E	03.6.0000E	03.7.0000E	03.8.0000E	03.9.0000E	03.1.0000E	04.802
61.2000E	04.1.4000E	04.1.6000E	04.1.8000E	04.2.0000E	04.2.2000E	04.802
72.4000E	03.4.0000E	04.2.8000E	04.3.0000E	04.1.0000E	03.2.0000E	03.802
83.0000E	03.4.0000E	03.5.0000E	03.6.0000E	03.7.0000E	03.8.0000E	03.802
99.0000E	03.1.0000E	04.1.2000E	04.1.4000E	04.1.6000E	04.1.8000E	04.802
\$2.0000E	04.2.2000E	04.2.4000E	04.2.6000E	04.2.8000E	04.3.0000E	04.802
\$1.0000E	03.8.0000E	03.9.0000E	03.1.0000E	03.3.0000E	03.6.0000E	03.802
\$1.6000E	04.1.8000E	04.2.0000E	04.2.2000E	04.2.4000E	04.1.4000E	04.802
\$2.0000E	04.3.0000E	04.1.0000E	03.2.0000E	03.3.0000E	03.4.0000E	03.802
\$5.0000E	03.6.0000E	03.7.0000E	03.8.0000E	03.9.0000E	03.1.0000E	04.802
\$2.4000E	04.2.6000E	04.1.6000E	04.1.8000E	04.2.0000E	04.2.2000E	04.802
DATA TL5/						
1 70*0.0.						BD2
23.0000E	03.4.0000E	03.5.0000E	03.6.0000E	03.7.0000E	03.8.0000E	03.802
39.0000E	03.1.0000E	04.1.2000E	04.1.4000E	04.1.6000E	04.1.8000E	03.802
42.0000E	04.2.2000E	04.2.4000E	04.2.6000E	04.2.8000E	04.3.0000E	04.802
5 10*0.0.						BD2
DATA TL6/						
11.0000E	01.2.0000E	01.3.0000E	01.4.0000E	01.5.0000E	01.6.0000E	01.802
27.0000E	01.8.0000E	01.9.0000E	01.1.0000E	01.2.0000E	01.3.0000E	00.802
31.6000E	00.1.8000E	00.2.0000E	00.2.5000E	00.3.0000E	00.3.5000E	00.802
44.0000E	00.5.0000E	00.6.0000E	00.7.0000E	00.8.0000E	00.9.0000E	00.802
51.0000E	01.1.2000E	01.1.4000E	01.1.6000E	01.1.8000E	01.2.0000E	01.802
62.5000E	01.3.0000E	01.3.5000E	01.4.0000E	01.5.0000E	01.7.5000E	01.802
71.0000E	02.63*0.0					BD2
DATA TL11/						
11.8756E	01.1.9016E	01.1.9259E	01.1.9484E	01.1.9693E	01.1.9888E	01.802
22.0070E	01.2.0229E	01.2.0397E	01.2.0545E	01.2.0684E	01.2.0815E	01.802
32.0937E	01.2.1053E	01.2.1162E	01.2.1265E	01.2.1362E	01.2.1455E	01.802
42.1543E	01.2.1626E	01.2.1705E	01.2.1781E	01.2.1853E	01.2.1988E	01.802
52.2111E	01.2.2224E	01.2.2329E	01.2.2425E	01.2.2515E	01.2.2598E	01.802
62.2676E	01.2.2749E	01.2.2817E	01.2.2881E	01.2.2941E	01.2.2998E	01.802
72.3052E	01.2.3151E	01.2.3240E	01.2.3320E	01.2.3394E	01.2.3461E	01.802
82.3522E	01.2.3579E	01.2.3631E	01.2.3680E	01.2.3725E	01.2.3767E	01.802
92.3806E	01.2.3825E	01.2.3867E	01.2.3907E	01.2.3945E	01.2.3980E	01.802
\$1.2240E	01.1.1570E	01.1.1570E	01.1.1570E	01.1.1570E	01.1.1570E	01.802
\$9.2600E	00.8.8000E	00.8.4100E	00.8.0800E	00.7.7900E	00.7.5200E	00.802
\$7.2900E	00.7.0900E	00.6.9000E	00.6.7300E	00.6.5700E	00.6.4200E	00.802
DATA TL12/						
2.5310E	01.2.1490E	01.1.9160E	01.1.7580E	01.1.6410E	01.1.5480E	01.802
21.4720E	01.1.4070E	01.1.3510E	01.1.3030E	01.1.2200E	01.1.1530E	01.802
31.0930E	01.1.0500E	01.1.0130E	01.1.0130E	00.9.4800E	00.9.2200E	00.802
43.9700E	00.8.7500E	00.2.2480E	01.1.7340E	01.1.4980E	01.1.3660E	01.802
51.2740E	00.1.2030E	01.1.1450E	01.1.0970E	01.1.0560E	01.1.0200E	01.802
69.6000E	00.9.1100E	00.8.7200E	00.8.4000E	00.8.0900E	00.7.8400E	00.802
77.6100E	00.7.4200E	00.7.2200E	00.7.0400E	00.6.1640E	01.2.7630E	01.802
82.4650E	01.2.2420E	01.2.0730E	01.1.9440E	01.1.8420E	01.1.7580E	01.802
91.6780E	01.1.6330E	01.1.5420E	01.1.4850E	01.1.4300E	01.1.3740E	01.802
\$1.3240E	01.1.2780E	01.1.2390E	01.1.2050E	01.1.1730E	01.1.1440E	01.802
\$2.7860E	01.2.3160E	01.2.0610E	01.1.8900E	01.1.7610E	01.1.6610E	01.802
\$1.5780E	01.1.5080E	01.1.4470E	01.1.3940E	01.1.3050E	01.1.2320E	01.802

1.1790E	01.1.1360E	01.1.0940E	01.1.0600E	01.1.0290E	01.1.0030E	01.8D2	386
\$9.7700E	00.9.5100E	00.2.9080E	01.2.6300E	01.2.4780E	01.2.3670E	01.8D2	387
\$2.2720E	01.2.1900E	01.2.1180E	01.2.0510E	01.1.9790E	01.1.9130E	01.8D2	388
\$1.8050E	01.1.7150E	01.1.6420E	01.1.5920E	01.1.5240E	01.1.4770E	01.8D2	389
\$1.4340E	01.1.3970E	01.1.3600E	01.1.3250E	01.1.2930E	01.1.2270E	01.8D2	390
DATA TL13/						01.8D2	391
11.9460E	01.1.6560E	01.1.4910E	01.1.3780E	01.1.2930E	01.1.2270E	01.8D2	392
21.1710E	01.1.1230E	01.1.0480E	01.1.0480E	01.9.8900E	00.9.4100E	00.8D2	393
39.0100E	00.7.6800E	00.8.3600E	00.8.1000E	00.7.8700E	00.7.6700E	00.8D2	394
47.4600E	00.7.2700E	00.2.3590E	01.2.0320E	01.1.8520E	01.1.7290E	00.8D2	395
51.6370E	01.1.5630E	01.1.5030E	01.1.4510E	01.1.4060E	01.1.3670E	01.8D2	396
61.3000E	01.1.2440E	01.1.1910E	01.1.1480E	01.1.1050E	01.1.0710E	01.8D2	397
71.0400E	01.1.0130E	01.9.8700E	00.9.6100E	00.2.7880E	01.2.5270E	01.8D2	398
82.3900E	01.2.2870E	01.2.2030E	01.2.1200E	01.2.0440E	01.1.9810E	01.8D2	399
91.9000E	01.1.8780E	01.1.7960E	01.1.7300E	01.1.6560E	01.1.5950E	01.8D2	400
\$1.5360E	01.1.4890E	01.1.4460E	01.1.4090E	01.1.3720E	01.1.3360E	01.8D2	401
\$6.0000E	00.8.2000E	00.9.6000E	00.1.0700E	01.1.1900E	01.1.3100E	01.8D2	402
\$1.4300E	01.1.5200E	01.1.6400E	01.1.7600E	01.1.9700E	01.2.1000E	01.8D2	403
\$2.0000E	01.1.5100E	01.1.3000E	01.1.1900E	01.1.1500E	01.1.1200E	01.8D2	404
\$7.1000E	01.1.1000E	01.2.6000E	01.1.3700E	01.1.0900E	01.1.1000E	00.8D2	405
\$7.7000E	00.5.7000E	00.6.0000E	00.5.4000E	00.4.9000E	00.4.7000E	00.8D2	406
\$4.7000E	00.5.2000E	00.5.7000E	00.6.1000E	00.6.4000E	00.6.6000E	00.8D2	407
\$6.9000E	00.6.5000E	00.7.0000E	00.7.0000E	00.7.0000E	00.7.0000E	00.8D2	408
DATA TL14/						01.8D2	409
13.6000E	00.5.0000E	00.6.6000E	00.8.4000E	00.1.0000E	01.1.1000E	01.8D2	410
21.1400E	01.1.1000E	01.1.0300E	01.1.0100E	01.1.0100E	01.1.0100E	01.8D2	411
31.0100E	01.1.0100E	01.5.3000E	00.7.7000E	00.8.8000E	00.9.5000E	00.8D2	412
59.9000E	00.1.0300E	01.1.0600E	01.1.0900E	01.1.1200E	01.1.1500E	01.8D2	413
61.2300E	01.1.3000E	01.1.4600E	01.1.6100E	01.1.8100E	01.2.0000E	01.8D2	414
72.2000E	01.2.4000E	01.2.6000E	01.2.8000E	01.1.0100E	00.1.1400E	00.8D2	415
81.3200E	00.1.5000E	00.1.6700E	00.1.8500E	00.2.0200E	00.2.2400E	00.8D2	416
92.4600E	00.2.6400E	00.3.0500E	00.3.4300E	00.3.7800E	00.4.0900E	00.8D2	417
\$2.4000E	00.4.5800E	00.4.7500E	00.4.8900E	00.4.9700E	00.5.0600E	00.8D2	418
\$2.7000E	00.3.0000E	00.3.5000E	00.4.1000E	00.4.8000E	00.5.7000E	00.8D2	419
\$3.7000E	00.7.6000E	00.8.3000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8D2	420
\$9.5000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8D2	421
\$9.5000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8.5000E	00.8D2	422
\$5.0000E	01.8.0000E	01.8.0000E	01.8.0000E	01.3.0000E	01.4.0000E	01.8D2	423
\$1.7000E	00.2.0000E	00.2.3000E	00.2.6000E	00.2.9000E	00.3.2000E	00.8D2	424
\$3.5000E	00.3.8000E	00.4.1000E	00.4.4000E	00.4.7000E	00.5.0000E	00.8D2	425
DATA TL15/						01.8D2	426
1.70*0.0.	00.6.2700E	00.6.3300E	00.6.3700E	00.6.4000E	00.6.4200E	00.8D2	427
25.2000E	00.6.4600E	00.6.4900E	00.6.5100E	00.6.5300E	00.6.5500E	00.8D2	428
35.4400E	00.6.5900E	00.6.6100E	00.6.6400E	00.6.6700E	00.6.7000E	00.8D2	429
5.10*0.0.						00.8D2	430
DATA TL16/						01.8D2	431
14.0079E	00.3.1300E	00.2.6494E	00.2.3144E	00.2.0661E	00.1.8767E	00.8D2	432
31.7293E	00.1.6122E	00.1.5175E	00.1.4398E	00.1.3204E	00.1.2336E	00.8D2	433
31.1679E	00.1.1166E	00.1.0753E	00.1.0006E	00.9.5003E	01.9.311E	01.8D2	434
43.8453E	01.8.4277E	01.8.1287E	01.7.8976E	01.7.7111E	01.7.5553E	01.8D2	435
57.4222E	01.7.2022E	01.7.0254E	01.6.8776E	01.6.7511E	01.6.6405E	01.8D2	436
66.4136E	01.6.2350E	01.6.0882E	01.5.9640E	01.5.7626E	01.5.4146E	01.8D2	437
75.1803E	01.6.63*0.0					01.8D2	438
						01.8D2	439
						01.8D2	440

[illegible]

[illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE AESOLN (X)
LOGICAL ERR
ROUTINE FOR CALCULATING EFFECTIVE AREA RATIO AND ITS DERIVATIVE
CALL TO AESOLN(X) MUST BE PRECEDED BY CALL GEOMAR(X,S1,S2)
      X = AXIAL COORDINATE ALONG NOZZLE
      DELBL = POUNDARY LAYER DISPLACEMENT THICKNESS DIVIDED BY ROAES
      S1 = GEOMETRIC AREA RATIO (PROVIDED BY CALLING ROUTINE)
      S2 = DERIVATIVE OF S1 (PROVIDED BY THE CALLING ROUTINE)
      S2 IS RESET TO THE DERIVATIVE OF THE EFFECTIVE AREALS
      RATIO
COMMON /RL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
      SW,RO,JOIM,IPCINT
1 DIMENSION DELP(2),DELSIP(2)
COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),
      NSECTU(2),ISHAPE(12,2),NPROFL(2),PRFLS,NBL
COMMON /AEGEOM/ SORTA,S1,S2
COMMON /ERROR/ ERR
COMMON /NEQ/ OMDST(2),DDELBL(2)
IF (NPRFLS.EQ.2) GO TO 20
IF (JOIM.EQ.0) GO TO 10
SORTAE=(SORTA-DELBL(1))/OMDST(1)
S1=SQRTAE**2
S2=(S2/SORTA-2.*DDELBL(1))/OMDST(1)*SORTAE
GO TO 40
S1=(S1-DELBL(1))/OMDST(1)
S2=(S2-DDELHL(1))/CMDST(1)
GO TO 40
CALL GMAR3 (X,DYDX,DZDX,Y,Z)
IF (ERR) RETURN
DO 30 K=1,2
DELP(K)=RO*DDELBL(K)
DELSIP(K)=RO*(1.-OMDST(K))
DENOM=(RTHCM(1)-DELSIP(1))*(RTHCM(2)-DELSIP(2))
FAC1=Y-DELP(1)
FAC2 =DELP(2)
S1=F, 1*FAC2/DENOM
S2=(FAC2*(DYDX-DDELBL(1)*RO)+FAC1*(DZDX-DDELBL(2)*RO))/DENOM
RETURN
END

```

CCCCCCCCCCCC

10

20

30

40

```

1 SUBROUTINE AGSOLN (AE,DEL,UPDOWN,AG,X)
2 LOGICAL ERR, FIX
3 SOLUTION FOR THE GEOMETRIC AREA RATIO WHEN THE EFFECTIVE AREA
4 RATIO AND BOUNDARY LAYER DISPLACEMENT THICKNESS ARE GIVEN
5
6     AE = EFFECTIVE AREA RATIO
7     DEL = DISPLACEMENT THICKNESS
8     UPDOWN= 1. IF DOWNSTREAM SOLUTION IS DESIRED
9     -1. IF UPSTREAM SOLUTION IS DESIRED
10    AG = CCUMPUTED GEOMETRIC AREA RATIO
11
12    COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),
13    NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL
14    COMMON /NEO/ GMDST(2),DCELBL(2)
15    COMMON /ERROR/ ERR
16    COMMON /BL/ DEBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
17    SW,RO,JOIM,IPCINT
18    DIMENSION DEL(2),DELP(2),DSTP(2)
19    DATA RNAME /6HAGSOLN/
20    IF (NPRFLS.EQ.2) GO TO 20
21    NOT A CHANNEL
22    IF (JDIM.NE.0) GO TO 10
23    TWO-DIMENSIONAL FLOW
24    AG=DEL(1)+OMDST(1)*AE
25    GO TO 100
26    AXISYMMETRIC FLOW
27    SORTA=DEL(1)+OMDST(1)*SORT(AE)
28    AG= SORTA**2
29    GO TO 100
30    CHANNEL
31    DO 30 L=1,2
32    DSTP(L)=RO*(1.-CMDST(L))
33    DELP(L)=RO*DEL(L)
34    FIX=.FALSE.
35    Y2ZC=RTHCM(1)*RTHCM(2)
36    AEC=(1.-DSTP(1)/RTHCM(1))*(1.-DSTP(2)/RTHCM(2))*AE-DELP(1)*DELP(2)
37    1/YOZO
38    A=AE
39    ICOUNT=0
40    CALL FINDX (A,UPDOWN,X)
41    IF (ERR) RETURN
42    ICOUNT=ICOUNT+1
43    IF (ICOUNT.GT.20) GO TO 90
44    CALL GVAR2 (X,Y,Z)
45    IF (ICOUNT.GT.1) GO TO 70
46    A=AEC+(Y*DELP(2)+Z*DELP(1))/YOZO
47    XO=X
48    FO=A-AE
49    IF (A.GE.1.) GO TO 50
50    X=0.
51    GO TO 60
52    CALL GEOMAR (X,A,DA)
53    IF (ERR) RETURN
54    F=AEC+(Y*DELP(2)+Z*DELP(1))/YOZO-A
55    IF (ABS(F)/A.LE.1.E-5) GO TO 120

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

C      IF (F.EQ.FO) GO TO 120
      XL=X
      X=X-F*(X-XO)/(F-FO)
      XO=XL
      FO=F
      IF (X*UPDOWN.GE.O.) GO TO 60
      SIGN OF X IS INCONSISTENT WITH UPDOWN
      IF (FIX) GO TO 90
      IF (AE.GT.1.001) GO TO 90
      TROUBLE IS DIAGNOSED AS DUE TO SONIC POINT TOO FAR DOWNSTREAM
      OF GEOMETRIC THROAT. TRY FIX -- RESET DISPLACEMENT THICKNESSES
      TO THEIR VALUES AT THE THROAT.
      DO 80 L=1,2
      DPLP(L)=DSTP(L)
      FIX=.TRUE.
      WRITE (6,140)
      GO TO 40
      80  WRITE (6,150) AE,DEL,UPDCWN,X
      NAMELIST /AGDMP/ OMDST,RO,DSTP,DELP,YOZO,RTHCM,AEC,A, AO,XO,F,FO,
      ;
      WRITE (6,AGDMP)
      CALL DUMP (RNAME)
      GO TO 130
      100 IF (AG.GT.1.) GO TO 110
      AG=1.
      X=0.
      GO TO 130
      110 CALL FINDX (AG,UPDOWN,X)
      GO TO 130
      120 AG=A
      130 RETURN
      C
      C
      C
      140 FORMAT (29H0+++++ FIX REQUIRED IN AGSOLN)
      150 FORMAT (30H0CONVERGENCE FAILURE IN AGSOLN,9X,3HAE=,1PE12.5,8X,4HDE
      1L=.2E12,5.8X,7HUPDOWN=.0PF4,1.8X,2HX=.1PE12.5)
      END

```

AGS 56
 AGS 57
 AGS 58
 AGS 59
 AGS 60
 AGS 61
 AGS 62
 AGS 63
 AGS 64
 AGS 65
 AGS 66
 AGS 67
 AGS 68
 AGS 69
 AGS 70
 AGS 71
 AGS 72
 AGS 73
 AGS 74
 AGS 75
 AGS 76
 AGS 77
 AGS 78
 AGS 79
 AGS 80
 AGS 81
 AGS 82
 AGS 83
 AGS 84
 AGS 85
 AGS 86
 AGS 87
 AGS 88
 AGS 89
 AGS 90
 AGS 91
 AGS 92
 AGS 93-

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

X S X S X S X S X S X S X S X S X S X S X S X S X S X S X S

```

SUBROUTINE AXSECT (LL,NN)
SUBROUTINE AXSECT ADDS DATA FOR UNSPECIFIED SPECIES PAIRS TO STEP
ALL OF THE EDITED CROSS SECTION DATA. SPECIES TO BE CONSIDERED ARE
GIVEN IN NUMERIC ORDER IN THE FIRST NN ENTRIES OF THE ARRAY I.
COMMON /TRANS1/ I,Q(3,20,20),ZM2(20)
COMMON /TRANS2/ KQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)
COMMON /TRANS3/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKG
IF (NN.LE.0) GO TO 40
LQ1=NQ(LL)+1
N=0
DO 20 IV=1,NN
II=I(IV)
DO 20 JV=IV,NN
JJ=I(JV)
IF (Q(1,II,JJ).NE.0.) GO TO 20
Q(1,II,JJ)=1.
NN=NN+1
LQ=400
DO 10 L=LQ1,399
LQ=LQ-1
IQ(LQ+1)=IQ(LQ)
JQ(LQ+1)=JQ(LQ)
IQ(LQ1)=I
JQ(LQ1)=JJ
CONTINUE
DO 30 L=LL,NKG
NQ(LL)=NQ(L)+N
RETURN
END

```

04

-407-


```

COMMON /SS/ CAS,US
COMMON /SWITCH/ XPB,DXPB,W
EQUIVALENCE (XP,XPB). (DXP,DXPB)
COMMON /DRV/ SHJGGJ
COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE
COMMON /BLRAD/ Y070
COMMON /MODP2/ MODLPT
COMMON /COLDSP/ CGMW,QPJ(10),IJCS(10),NCS
COMMON /TNONEQ/ CTE,DTE,BPAR,EPAR(2.25),NT,ITR(25),KTF(25),
1 KTR(25),ICH(20),IPA(25)
DATA ONE /1./, ZERO /0./
NAMELIST /OMP/ IPOINT,X,T,P,RH,U,AM,H,AE,AG,AJ,UPRIME,AS,DX,HE,HR,
1 HREF,HW,TE,TREF,COSB,DADX,YZ,DYZDX,VISC,VISCR,VISCV,
2 PR,PRREF,PRW,ALPR,ORDIN,XI,OMEGA,SIGH,DMDX,DLM,US,RO,
3 SW,NBL,ITYPE,RER,BLINT,ENFRIM,SN,XSN,THETA1,THETA,HTR,
4 DELBL,DDELBL,XL,RAF,TAUW,CWDOT,XTEST,GAMMAE,WW,AVCON,B,EK1,
5 OLDMF,TRCALC,TEP,TREFP,RMR,FO,F2
DELBL=DISPLACEMENT THICKNESS
THETA=MOMENTUM THICKNESS
THETA1=MOMENTUM THICKNESS FOR TWO-DIMENSIONAL FLOW
THE BOUNDARY LAYER THICKNESSES DELBL, THETA, THETA1 ARE ALL
NONDIMENSIONALIZED USING THE GEOMETRIC THROAT RADIUS RO. THEY
ARE CONVERTED TO INCH UNITS FOR OUTPUT, IN SUBROUTINE OUT1.
IF (X,GT,XP,OR,IPOINT,EQ,1) GO TO 10
IF (X,EQ,CXR) GO TO 550
WRITE (6,CMP) X,XP
GO TO 550
10 DO 540 L=1,NPRFLS
IF (IPOINT,NE,1,OR,L,EQ,2) GO TO 90
SWP1=SW+1.
B=2.14+1.28*SWP1+0.93*(SWP1**2)**2
3M1=B-1.
EKL AND B ARE THE COEFFICIENTS IN THE COHEN-RESHOTKO LINEAR FIT TO BLA
THE MOMENTUM PARAMETER (CAP.N) AS A FUNCTION OF THE CORRELATION
NUMBER SN (SMALL N). THE FORMULA IS N=EK1+B*SN.
SVFAC=SWP1*(7.+3.4*SWP1)
SWFAC2=1.+4.5*SWP1**0.9
HCON=CRP/CMA
HW=CPWALL*TWALL
TEQ=0.
TREFP=0.
H0=CHA#HCON
DO 20 I=1,ISS
XSAVE(I)=SAVEC(I)
XTEST(I)=-1.
DO 40 I=1,ISS
DO 20 K=1,NCS
IF (IJCS(K),NE,I) GO TO 30
SAVEC(I)=QPJ(K)
GO TO 40
CONTINUE
SAVEC(I)=0.
CONTINUE
CALL TRANSP (TWALL,PRESA)

```

```

150 PRW=PR
150 VISCW=VISC
150 RWMW=VISCW*PRESA*CGMW/82.05/TWALL/RHAP
150 DO 50 I=1,ISS
150 SAVEC(I)=XSAVE(I)
150 IF (NPRFLS.EQ.1) GO TO 60
150 Y0Z0=RTHCM(1)*RTHCM(2)
150 ITYPE=3
150 GO TO 80
150 IF (JDIM.EQ.0) GO TO 70
150 ITYPE=2
150 GO TO 80
150 ITYPE=1
150 ITYPE IS A FLAG FOR TYPE OF NOZZLE GEOMETRY
150 ITYPE=1 TWO-DIMENSIONAL
150 ITYPE=2 AXISYMMETRIC
150 ITYPE=3 RECTANGULAR CHANNEL
150 AVCON=W/R0
150 R0 IS SET IN MAIN
150 COUPLD=.FALSE.
150 ISMD=30
150 DDELBL(1)=0.
150 DDELBL(2)=0.
150 IF (L.EQ.2) GO TO 100
150 IF (N1.EQ.2) H0=CHA*HCON
150 AS=SQRT(CAS*CT)
150 IF (FINAL) GO TO 110
150 B-INTI=ELINT(L)
150 XII=XI(L)
150 IF (L.EQ.2) GO TO 220
150 TE=CTAP*T
150 DO 120 I=1,ISS
150 IF (AH5(XTEST(I)-SAVEC(I)).GT.0.01) GO TO 130
150 CONTINUE
150 CLDWF=.TRUE.
150 TRCALC=.FALSE.
150 GO TO 150
150 CLDWF=.FALSE.
150 DO 140 I=1,ISS
150 XTEST(I)=SAVEC(I)
150 GO TO 160
150 IF (ABS(TE-TEP)/TE.LT.0.005) GO TO 180
150 CALL TRANSP (TE,P*PRESA)
150 TEP=TE
150 IF (ERR) RETURN
150 IF (IPOINT.LE.1) PRREF=PR
150 DO 170 I=1,4
150 TRPSV(I)=TRP(I)
150 HE=HCON*H
150 PRPW=PRREF*.56
150 HR=HE+PRPW*(H0-HE)
150 HR=RECOVERY ENTHALPY
150 HREF=.5*(HF+HW)+.22*(HR-HE)
150 TREF=CTAP*HREF/H0
150 CALL GEOMAP (X,AG,CADAX)

```

ORIGINAL PAGE IS
OF POOR QUALITY

BLA 165
BLA 166
BLA 167
BLA 168
BLA 169
BLA 170
BLA 171
BLA 172
BLA 173
BLA 174
BLA 175
BLA 176
BLA 177
BLA 178
BLA 179
BLA 180
BLA 181
BLA 182
BLA 183
BLA 184
BLA 185
BLA 186
BLA 187
BLA 188
BLA 189
BLA 190
BLA 191
BLA 192
BLA 193
BLA 194
BLA 195
BLA 196
BLA 197
BLA 198
BLA 199
BLA 200
BLA 201
BLA 202
BLA 203
BLA 204
BLA 205
BLA 206
BLA 207
BLA 208
BLA 209
BLA 210
BLA 211
BLA 212
BLA 213
BLA 214
BLA 215
BLA 216
BLA 217
BLA 218
BLA 219

```

IF (ERR) RETURN
AG=GEOMETRIC AREA RATIO (AREA DIVIDED BY AREA AT THROAT)
AE=EFFECTIVE AREA RATIO ALLOWING FOR BOUNDARY LAYER DISPLACEMENT
THICKNESS
GO TO (190,200,210), ITYPE
AJ=1.
COSB(1)=1./SQRT(1.+(RO*DADX)**2)
GO TO 230
AJ=AG
COSB(1)=1./SQRT(1.+0.25*(RO*DADX)**2/AG)
GO TO 230
CALL GMAR3 (X,DYZDX(1),DYZDX(2),YZ(1),YZ(2))
LPR=LPRIME(L)
AJ=(YZ(LPR))**2
COSB(L)=1./SQRT(1.+(DYZDX(L))**2)
COSB=COSINE OF THE ANGLE BETWEEN THE NOZZLE WALL AND THE AXIS OR
CENTER PLANE.
IF (L.EQ.2) GO TO 270
UPRIME=U*US
IF (OLDMF.AND.ABS(TREF-TREFP)/TREF.LT.0.005) GO TO 240
CALL TRANSP (TREF,P*PRESA)
IF (ERR) RETURN
TREFP=TREF
TRCALC=.TRUE.
PRREF=PR
VISCR=VISC
IF (.NOT.TRCALC) GO TO 250
VR=VISCR/VISCW
TREF=TREF/TWALL
OMEGA=ALOG(VR)/ALOG(TR)
IF (IPOINT.NE.1) GO TO 250
EK1=0.38-0.76*PRREF*(1.-CMEGA)*EXP(-6.67*SWP1)
IF (EK1.GE.0.01) GO TO 250
WRITE (6,DMP)
EK1=0.01
DO 260 I=1,4
TRP(I)=TRPSV(I)
AM=UPRIME/AS
ORDIN(L)=AJ*AM**BM1*AS*RH*T
IF (ITYPE.EQ.2) R=SQRT(AG)
R=RATIO OF LOCAL NOZZLE RADIUS TO THROAT RADIUS.
IF (IPOINT.NE.1) GO TO 290
FIRST PCINT
DX=(X-XZERO)/10.
DXSQ=DX**2
XX=XZERO
XI(L)=0.
BLINT(L)=0.
CC=AM*AG
CALL RADII (ITYPE,XZERO,R1,AG1,AGJ,L)
AVI=CC/AG1
ORD1=AGJ*AM1**BM1
DO 280 I=1,10
XX=XX+DX

```

```

280 CALL RADIUS (ITYPE,XX,R2,AG1,AGJ,L)
    DXI=SQRT(DXSQ+(R2-R1)**2)
    XI(L)=XI(L)+DXI
    AM1=CC/AG1
    ORD2=AGJ*AM1**BM1
    BLINT(L)=BLINT(L)+0.5*(ORD1+ORD2)*DXI
    ORD1=ORD2
    R1=R2
    BLINT(L)=BLINT(L)*AS
    DLM(L)=-DADX/AG
    X IS THE AXIAL COORDINATE DOWN THE NOZZLE, ZERO AT THE GEOMETRIC
    C THROAT.
    C XI IS THE STREAMWISE COORDINATE IN THE BOUNDARY LAYER, ZERO AT THE
    C NOZZLE ENTRANCE (WHERE X=XZERO).
    GO TO 370
290 DX=X-XP
    IF (INEQ.EQ.0) GO TO 320
    C COMPUTE DERIVATIVES FROM RATE DATA
    IF (NT.EQ.1) GO TO 300
    DUDX=GJ(1)*CCPJ(1)*DTE+SCPGH*DT
    GO TO 310
300 DUDX=SCPG*DT
310 DUDX=-CWA/U*(DUDX+SHJDGJ)
    IF (NT.EQ.2) DUDX=DUDX+DCHA/SU
    DLM(L)=DUDX/U-0.5*DT/T
    GO TO 360
320 IF (IPQINT.GT.2) GO TO 340
330 DMDX=(AM-AMP)/DX
    GO TO 350
    C CALCULATE DERIVATIVE FROM QUADRATIC FIT
340 IF (DX.LE.0.) GO TO 370
    DX2=DX+DXP
    XPXP=X+XP
    DMDX=(AM-AMP)/DX
    C2=(DMDX-(AM-AMPP)/DX2)/DXP
    C1=DMDX-C2*XPXP
    IF (DMDX.LT.0.) GO TO 330
    DLM(L)=DMDX/AM
    C TRAPEZOIDAL RULE INTEGRATION
350 DXI=DX/COSB(L)
360 XI(L)=XI(L)+DXI
    BLINT(L)=BLINT(L)+0.5*(ORDIN(L)+ORDINP(L))*DXI
    DLM(L)=DLM(L)*COSB(L)
    RER=RH*RHAP*UPRIME*R0/VISC
    RER=REYNOLDS NUMBER BASED ON THROAT RADIUS AND FREE-STREAM
    C CONDITIONS.
    ENPRIM=BLINT(L)/ORDIN(L)/R0*EK1
    RMR=RMW*P/VISC/RH
    THETA1=SQRT(ENPRIM/RER*MR)
    GO TO (380,390,380), ITYPE
    C THETA(L)=THETA1
    GO TO 400
380 TRANSVERSE CURVATURE CORRECTION FOR AXISYMMETRIC NOZZLES
    C SQT=SQRT(1.-2.*COSB(L)*THETA1/R)
390

```

ORIGINAL PAGE IS
 OF POOR QUALITY

```

400      THETA(L)=R/COSB(L)*(1.-SQT)
      IF (.NOT.FINAL.AND.IPOINT.NE.1) GO TO 430
      SN(L)=-ENPRIM*RO*DLM(L)
      IF (IPOINT.NE.1) GO TO 410
      XSN(L)=SN(L)
      GO TO 420
410      W=EXP(-AVCON*DX/AG)
      XSN(L)=(1.-W)*SN(L)+W*XSN(L)
420      XSNV=AMIN1(XSN(L),ZERO)
      ALPR=ALOG(PRREF)
      GAMMAE=AS*2/(8.3144E7/CM*TE)
      IF (GAMMAE.LE.1.) GAMMAE=1.01
      SIGH=1./(1.+2./(GAMMAE-1.)/AM**2)
      HTR(L)=0.75*(EXP(4.6*XSNV))*(1.+(3.-15.6*XSNV)*SWP1)-1.)
      HTRP1(L)=(HTR(L)+1.)*(SWP1*(-0.4*ALPR)-0.47*SIGH*ALPR)
      HTRP1=HTRP1(L)-1.
      HFPI=HTRP1(L)/T
      HF=HFPI-1.
      DELBL(L)=HF*THETA(L)
      IF (IPOINT.LE.1.OR..NOT.FINAL) GO TO 470
      IF (COUPLD) GO TO 440
      IF (IUPD.EQ.1) GO TO 470
      COUPLD=.TRUE.
440      DDELBL(L)=(DELBL(L)-DELBLP(L))/(X-XP)
      IF (ISMD.EQ.1.OR.L.EQ.2) GO TO 450
      WD=1./ISMD
      GO TO 460
450      IF (L.EQ.2) GO TO 460
      WD=0.5*AMIN1(ONE.(DX/DXO)*2)
460      DDELBL(L)=WD*DDELBL(L)+(1.-WD)*DDELBLP(L)
      IF (ISMD.EQ.1.OR.L.LT.NPRFLS) GO TO 470
      ISMD=ISMD-1
      CONTINUE
470      IF (FINAL) GO TO 480
      BLINT(L)=BLINTI
      XI(L)=XI
      GO TO 540
480      DELBLP(L)=DELBL(L)
      ORDINP(L)=ORDIN(L)
      OM1=AMAX1(1.-OMEGA,ZERO)
      F3=EXP(-0.427*(SWP1*(-OM1)-1.))
      AX=15.5*EXP(-29.*SWP1)+0.67*EXP(4.37*SWP1)
      GX=0.7+0.47*SWP1
      F1=0.76*OM1**1.2*(1.-SORT(1.-SIGH))
      IF (SN(L).LT.-1.E-20) F1=F1/(1.+AX*(-SN(L))**BX)
      IF (FC) 490.490.500
      F3=0.
490      GO TO 510
500      F3=EXP(3.5*AMIN1(SN(L),ZERO)/F0**1.57)
510      F2=F0+0.45*OM1*(1.-F3)+F1
      IF (SN(L).LE.-1.E-20) GO TO 520
      XL=C.2205
      GO TO 530
520      XL=0.2205+SWFAC2/(-2.7/SN(L)+5.06/(-SN(L))**0.224)
530      XL=F2*XL

```

```

BLA 275
BLA 276
BLA 277
BLA 278
BLA 279
BLA 280
BLA 281
BLA 282
BLA 283
BLA 284
BLA 285
BLA 286
BLA 287
BLA 288
BLA 289
BLA 290
BLA 291
BLA 292
BLA 293
BLA 294
BLA 295
BLA 296
BLA 297
BLA 298
BLA 299
BLA 300
BLA 301
BLA 302
BLA 303
BLA 304
BLA 305
BLA 306
BLA 307
BLA 308
BLA 309
BLA 310
BLA 311
BLA 312
BLA 313
BLA 314
BLA 315
BLA 316
BLA 317
BLA 318
BLA 319
BLA 320
BLA 321
BLA 322
BLA 323
BLA 324
BLA 325
BLA 326
BLA 327
BLA 328
BLA 329

```

```

TAUW(L)=0.00209**XL*RMV*VISC*UPRIME/THETA(L)/RO
RA=SWP1*(-0.86*OMEGA)*(1./(6.2+2.35/SWP1**0.86))+(1.--PRREF)**2*(0.86)
1185*(1.-2.*SWP1)+0.29*SWP1**0.38*SIGH/OMEGA))
R9=0.94-1.31*SWP1**0.486-ALPR*(0.62*SWP1+(3.5*OM1-0.49-3.61*SWP1)*BLA
1SIGH)
RAF=PRREF*(-0.35)*(1.-SN(L)/(RA-RB*SN(L)))
QWDOT(L)=1760.*TAUW(L)*(HR-HW)/UPRIME/PRW/RAF
DDBLP(L)=DDELBL(L)
IF (L.LT.NPRFLS) GO TO 540
IPOINT=IPOINT+1
AMP=AMP
XP=X
DXP=DX
IF (.NOT.MODLPT) DXO=DX
REFP=REF/RO*30.48
CONTINUE
IF (ISW48.GT.0) WRITE (6,DMP)
IF (ISW48.LT.0) WRITE (6,570) X,IPOINT,ISMD,WD,(DDELBL(J),J=1,NPRF
1LS)
RETURN
FORMAT (36H0ZERO OR NEGATIVE STEP IN BLAYER. X=.1PE15.8,10X,3HXP=,BLA
1E15.9)
FORMAT (12H BLAYER X=.F10.5,5X,7HIPOINT=.14,5X,5HISMD=,12,5X,3HWBLA
1D=.F6.4,5X,7H0LELBL=.2F15.6)
FCRMT (56H0**** CALCULATED EK1 LESS THAN 0.01. EK1 RESET TO 0.01BLA
1.)
END

```

540

550

C

C

C

560

570

580

1.

END

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE BLCALL (FINAL)
CALLING ROUTINE FOR BOUNDARY LAYER IN NCNEEQUILIBRIUM CALCULATIONS
LOGICAL FINAL
REAL ACOM(30),ELMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFNX,AMACH,
COMMON C,CARB,CST,CH,CTA,CTT,CX,CXB,CXMAX,DATEST,DBTEST,DELT1,
1 CCRB,CRS,CTT,CX,CXB,CXMAX,DATEST,DBTEST,DELT1,
2 CTP,CTPL,CST,CH,CTA,CTT,CX,CXB,CXMAX,DATEST,DBTEST,DELT1,
3 DELT2,DELTAX,DLOGA,DLOGR,PRESTH,PRHAP,ROBARP,ROB,
4 DELT2,DELTAX,DLOGA,DLOGR,PRESTH,PRHAP,ROBARP,ROB,
5 SPTTEST,PRESA,RHOC,RHOP,SL,SL64,SU,SU2,SUMG,
6 RHOBAR,RHOC,RHOP,SL,SL64,SU,SU2,SUMG,
7 SEN,SHPG,SC,SL,SL64,SU,SU2,SUMG,
8 TEST,TESTB,TPRINT,STOP,ZP,
COMMON BE(64),BET(20),BLBK(31),CAI(64),CAPXTH(20),
1 CCPJ(20),CEACT(64),CGI(20),CGMU(20),CHII(20),
2 CLNIMC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),GJA(20),
3 GJ5(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),OM(20),
4 OUI(64),SAJ(20),SDCHI(64),SENT(20),SHJ(20),SHJA(20),
5 SKIL(20),SS(20),TB(30),TFA(20),TFB(20),TFC(20),
6 TFD(20),TFE(20),TFK(20),THEV(20),XMJAT(20),XNUI(64),
COMMON BETA(64,20),ELJ(10,20),GELJ(10,20),
1 XNUIJ(64,20),XNUIJP(64,20),
COMMON IC,IM,INEQ,INEQV,IP,IRUN,ISC,ISCP1,
1 ISMC,ISMCK,ISR,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,
2 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,ISW5A,
3 ISW5B,ISW6A,ISW6B,IUPD,IZERO,JJK,LC,MI,NFIT,
4 NIT,NNN,NNS,NOS,NOT,NREST
COMMON IGJ(20),IGM(20),IT3(5),KUR(64,20),LPIJ(20,10)
COMMON ACOM,ELMENT,HP
DIMENSION AAA(22,24),BTA(64,20),CAPO(31),CCI(20),DGJ(20),SBR
1J(20),SDGJ(20),SHJAP(20),THEVP(20)
EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),
1 (BLHK(1),CAPO(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
COMMON /TEMPRY/SAVEC(20)
IF (INEQ.EQ.0) GO TO 20
DO 10 J=1,ISS
SAVEC(J)=GJ(J)*CM
GO TO 40
CT=CT+PCT
RHO=RHO+PRHO
DO 30 J=1,ISS
SAVEC(J)=(GJ(J)+PERTGJ(J))*CM
IF (ISW3B.EQ.0) GO TO 50
CALL BLAYER (FINAL)
IF (INEQ.NE.0) GO TO 60
CT=CT-PCT
RHO=RHO-PRHO
RETURN
END

```

1 BLC
2 BLC
3 BLC
4 BLC
5 BLC
6 BLC
7 BLC
8 BLC
9 BLC
10 BLC
11 BLC
12 BLC
13 BLC
14 BLC
15 BLC
16 BLC
17 BLC
18 BLC
19 BLC
20 BLC
21 BLC
22 BLC
23 BLC
24 BLC
25 BLC
26 BLC
27 BLC
28 BLC
29 BLC
30 BLC
31 BLC
32 BLC
33 BLC
34 BLC
35 BLC
36 BLC
37 BLC
38 BLC
39 BLC
40 BLC
41 BLC
42 BLC
43 BLC
44 BLC
45 BLC
46 BLC
47 BLC
48 BLC
49 BLC
50 BLC
51 BLC
52 BLC
53 BLC

C	SUBROUTINE BXSECT (L,MV,I,J)	1	BXS
C	SUBROUTINE BXSECT DETERMINES THE CORRESPONDENCE BETWEEN THE KQ AND BX	2	BXS
	V ARRAYS AND SEARCHES THE V ARRAY FOR REFERENCES TO SPECIES PAIRS.	3	BXS
	COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JO(400),NKQ	4	BXS
	COMMON /TRANS8/ NV(15),N	5	BXS
	IF (L.NE.0) GO TO 40	6	BXS
	MV=1	7	BXS
	MADD=0	8	BXS
10	L=L+1	9	BXS
	K=KQ(L)	10	BXS
	IF (K.EQ.9) GO TO 30	11	BXS
	IF (K.NE.12) GO TO 50	12	BXS
20	MADD=4	13	BXS
30	MADD=MADD-2	14	BXS
	MM=MV+MADD	15	BXS
	I=V(MM+1)	16	BXS
	J=V(MM+2)	17	BXS
	RETURN	18	BXS
40	V(MM+1)=I	19	BXS
	V(MM+2)=J	20	BXS
	IF (MADD.NE.0) GO TO 20	21	BXS
50	MM=MV+NV(K)	22	BXS
	IF (L.LT.NKQ) GO TO 10	23	BXS
	L=0	24	BXS
	RETURN	25	BXS
	END	26	BXS

```

SUBROUTINE COMM
LOGICAL ERR, FAILED, LOGC, NOREAC, RCALC
REAL ACOM(30), ELEMENT(10), HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22,24), CDIJ(20,10), CAPX(20)
COMMON A, AFNTS, AFNX, AMACH,
C, CARB, CH, CTA,
1 CRB, CRS, CSTA, CTT, CX, CXB,
2 CTP, CTPL, CTT, CX, CXB,
3 CTP, CTPL, CTT, CX, CXB,
4 DELT2, DELTAX, DLOGR, DLOGR, PRESTH, DT,
5 SPTTEST, PRES, PRES, RHOP, RHOP, RHOP, RHOP,
6 RHODAR, RHOC, RHOC, RHOP, RHOP, RHOP, RHOP,
7 SEN, SHPG, SHPG, SHPG, SHPG, SHPG, SHPG,
8 TEST, TESTB, TPRINT, TSTOP, TUP,
COMMON BE(64), BET(20), BLBK(31), CAI(64), CAPXTH(20),
1 CCPJ(20), CFACT(64), CGI(20), CGMU(20), CHI(64), CHI(20),
2 CLN1MC(64), CLNPI(64), CMW(20), ETAI(54), ETAI(20), GJA(20),
3 GJ(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), QM(20),
4 Q(64), SAJ(20), SDCHI(64), JENT(20), SHJ(20), SHJA(20),
5 SKIL(20), SS(20), TH(30), TFA(20), TFC(20),
6 TFD(20), TFE(20), TFK(20), THEV(20), XMJAT(20), XNUI(64),
COMMON BETA(64,20), ELJ(10,20), GELJ(10,20),
1 XNUIJ(64,20), XNUIJP(64,20),
COMMON IC, IM, INEQ, INFOV, IP, IRUN, ISC, ISCP1,
1 ISMC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
3 ISW5B, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, MI, NFIT,
4 NIT, NNN, NNS, NQS, NQT, NTEST,
COMMON IGJ(20), IGM(20), ITB(5), KUR(64,20), LPIJ(20,10),
COMMON ACOM, ELEMENT, HP
COMMON ACOM, AA(22,24), BTA(64,20), CAPQ(31), CCI(20), DGJ(20), GJ(20), SB
1 J(20), SDGJ(20), SHJAP(20), THEVP(20)
EQUIVALENCE (AA(1,1), AAA(1,1)), (BETA(1,1), BTA(1,1)),
1 (BLBK(1), CAPQ(1)), (CAPXTH(1), CCI(1)), (GJA(1), DGJ(1)),
2 (CAPX(1), GJ(1)), (SAJ(1), SBJ(1)), (SS(1), SDGJ(1)),
3 (SHJA(1), SHJAP(1)), (THEV(1), THEVP(1))
COMMON /ERROR/ ERR
COMMON /CERROR/ FAILED
COMMON /TONEG/ CTE, DTE, PPAR, EPAR(2,25), NT, ITR(25), KTF(25),
1 KTR(25), ICH(20), IPA(25)
COMMON /AREA/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12,2), NPROFL(2), NPREFL, NBL
COMMON /TNDG/ XMJATD(20,2), CLNTD(2), CTD(2)
COMMON /TNC/ SUMGH, SCPGH, QDPR, ODPE
COMMON /GLIM2/ NOREAC(64)
DATA VIC1, VIC2 /1.094E24, 1.1328E-16/, VNC /4.99E29/
DIMENSION ALGJ(20)
EQUIVALENCE (ALGJ(1), AA(1,24))
NAMELIST /COMDMR/ CX, AFNX, DLOGA, RHOD, DLOGR, SUMG, SHPG, RHAP, CRP,
1 SU2, CSTA, Z1, Z2F, CLNTD, GJ, CHI, PI, XMJATD, CTD, SUMGH, SCPGH,
2 QDPR, ODPE, RKR, RKF, UELAS, CLNPI, CLN1MC
DATA ICOUNT /0/, ONE /1./
DATA ICYCLE /90000000/
ICOUNT=ICOUNT+1
IF (ISW5B.LT.0) ICYCLE=-ISW5B

```

```

10 IF (ICYCLE.EQ.90000000) GO TO 30
11 IF (MOD(ICOUNT,ICYCLE)) 10,20,10
12 J TO 30
13 ISWB=1
14 WRITE (6,550) ICOUNT
15 CONTINUE
16 FAILED=.FALSE.
17 SUMG=0.
18 SCPGH=0.
19 QDPR=0.
20 QDPE=0.
21 DO 70 J=1,ISS
22 IF (GJ(J)) 80,40,50
23 ALGJ(J)=-1000.
24 GO TO 60
25 ALGJ(J)=ALOG(GJ(J))
26 IF (J.LT.NT) GO TO 70
27 SUMG=SUMG+GJ(J)
28 SCPGH=SCPGH+GJ(J)*CCPJ(J)
29 CONTINUE
30 SUMGH=SUMG
31 SCPG=SCPGH
32 IF (NT.EQ.1) GO TO 100
33 SUMG=SUMG+GJ(1)
34 SCPG=SCPG+GJ(1)*CCPJ(1)
35 TEP=CTE*CTAP
36 GO TO 100
37 WRITE (6,560)
38 FAILED=.TRUE.
39 RETURN
40 IF (INEQ) 110,150,110
41 SFN=0.0
42 SHPG=0.
43 DO 130 J=1,ISS
44 IF (GJ(J)) 80,130,120
45 SEN=SEN+CJ(J)*((SENT(J)-ALGJ(J))
46 SHPG=SHPG+GJ(J)*SHJ(J)
47 CM=1./SUMG
48 CH=CMA*SHPG
49 SU2=2.*((CHA-CH)
50 IF (SU2.GE.0.) GO TO 140
51 WRITE (6,570)
52 GO TO 90
53 SU=SU+(SU2)
54 CALL GECM
55 IF (ERR) RETURN
56 RHOP=RHOP+RHAP
57 RHPL=ALOG(RHOP)
58 Z1=ALOG(SL64/SU)+CSTA
59 Z2F=41.2918*CRP
60 RCALC=.FALSE.
61 MAIN LOC. OVER THE REACTIONS
62 DO 460 I=1,ISR
63 IF (INCREAC(I)) GO TO 450

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

COM 56
COM 57
COM 58
COM 59
COM 60
COM 61
COM 62
COM 63
COM 64
COM 65
COM 66
COM 67
COM 68
COM 69
COM 70
COM 71
COM 72
COM 73
COM 74
COM 75
COM 76
COM 77
COM 78
COM 79
COM 80
COM 81
COM 82
COM 83
COM 84
COM 85
COM 86
COM 87
COM 88
COM 89
COM 90
COM 91
COM 92
COM 93
COM 94
COM 95
COM 96
COM 97
COM 98
COM 99
COM 100
COM 101
COM 102
COM 103
COM 104
COM 105
COM 106
COM 107
COM 108
COM 109
COM 110

```

```

C      COMPUTATION OF PICHI
LOGC=.FALSE.
IF (NT.EQ.1) GO TO 160
KF=KTF(I)
KR=KTR(I)
GO TO 170
160  KF=1
     KR=1
170  IF (KF-3) 180,190,200
180  CLKF=ETAI(1)*CLNTD(KF)+CAI(1)-CEACT(1)/(CTD(KF)*CRP)
     GO TO 240
190  CLKF=ETAI(1)*CLNTD(2)+CAI(1)+ALOG(1.-EXP(-CEACT(1)/(CTD(1)*CRP)))
     GO TO 240
200  IF (RCALC) GO TO 220
     RCALC=.TRUE.
     IF (NPRFLS.EQ.2) GO TO 210
     CALL GMAR (CX,R)
     GO TO 220
210  CALL GMAR2 (CX,Y,Z)
     R=1./SORT(0.5*(1./Y**2+1./Z**2))
220  IF (KF.FQ.4) GO TO 230
     CLKF=CAI(1)-0.5*ALOG(R)
     GO TO 240
230  IPAI=IPA(I)
     TAU=SPAR*RHOP*GJ(IPAI)*R
     CLKF=ETAI(1)*CLNTD(2)-ALOG(AMAX1(ONE,TAU))+CAI(1)
240  CLNPI(I)=CLKF+Z1
     CLKR=CLKF
     CLNIMC(I)=0.0
     T6=0.0
     T7=0.0
     IF (KR.EQ.0) GO TO 250
     Z2=Z2F*CTD(KR)
     Z3=Z2*RHOP
     CLZ2=ALOG(Z2)
     CLZ3=ALOG(Z3)
250  DD 280 J=1,ISS
     IF (XNUIJ(I,J).EQ.0.0) GO TO 260
     T6=T6+XNUIJ(I,J)*ALGJ(J)
260  IF (KUR(I,J).EQ.0) GO TO 270
     T7=T7+GJ(J)
270  IF (BETA(I,J).EQ.0.0.OR.KR.EQ.0) GO TO 280
     CLNIMC(I)=CLNIMC(I)+BETA(I,J)*(ALGJ(J)+CLZ3+XMJATD(J,KR))
280  CONTINUE
     IF (KR.EQ.0) GO TO 300
     IF (CLNIMC(I).LT.20.) GO TO 290
     LOGC=.TRUE.
     GO TO 310
290  CHI(I)=1.0-EXP(CLNIMC(I))
     GO TO 310
300  CHI(I)=1.0
310  CLNPI(I)=CLNPI(I)+(XNUI(I)-1.0)*RHPL+T6
320  IF (QQ(I)) 320,350,320
330  CLTBF=ALOG(RHOP*T7)

```

```

COM 111
COM 112
COM 113
COM 114
COM 115
COM 116
COM 117
COM 118
COM 119
COM 120
COM 121
COM 122
COM 123
COM 124
COM 125
COM 126
COM 127
COM 128
COM 129
COM 130
COM 131
COM 132
COM 133
COM 134
COM 135
COM 136
COM 137
COM 138
COM 139
COM 140
COM 141
COM 142
COM 143
COM 144
COM 145
COM 146
COM 147
COM 148
COM 149
COM 150
COM 151
COM 152
COM 153
COM 154
COM 155
COM 156
COM 157
COM 158
COM 159
COM 160
COM 161
COM 162
COM 163
COM 164
COM 165

```

340 GO TO 350
 PI(I)=J.O
 CLTBF=-10000000.
 GO TO 370
 350 CLTBF=0.
 360 PI(I)=EXP(CLNPI(I)+CLTBF)
 370 IF (LOGC) GO TO 380
 PICH(I)=PI(I)*CHI(I)
 GO TO 390
 330 PICH(I)=-EXP(CLN1MC(I)+CLNPI(I)+CLTBF)
 390 IF (NT.EQ.1) GO TO 460
 C COMPUTATION OF RADIATIVE LOSS QDPR AND ENERGY TRANSFER TO THE
 C ELECTRON GAS QDPE
 SUMF=0.
 RKFE=EXP(CLKFE)
 SUMR=0.
 CALL EPART (I,TEP,EF,ER,QF,QR)
 DO 410 J=1,ISS
 IF (KR.EQ.0) GO TO 400
 CLKR=CLKR+RETA(I,J)*(CLZ2+XJATD(J,KR))
 400 SUMF=SUMF+XNUIJ(I,J)*(RHPL+ALGJ(J))
 410 SUMR=SUMR+XNUIJP(I,J)*(RHPL+ALGJ(J))
 IF (KR) 420,420,430
 420 RKRF=0.
 GO TO 440
 430 RKFE=EXP(CLKFE)
 440 SUMF=EXP(SUMF+CLTBF)
 SUMR=EXP(SUMR+CLTBF)
 QDPR=QDPR+QF*RKFE*SUMF-QR*RKRF*SUMR
 QDPE=QDPE+EF*RKFE*SUMF-ER*RKRF*SUMR
 NAMELIST /CDP/ I,KF,KR,CLKF,CLKR,T6,T7,22,Z3,CLZ2,CLZ3,
 1 CLTBF,RKFE,RKRF,EF,ER,QF,QR,SUMF,SUMR,QDPR,QDPE
 IF (INEQ.NE.0.AND.ISWEB.NE.0) WRITE (6,CDP)
 GO TO 460
 PICH(I)=0.
 CONTINUE
 IF (INEQ) 470,540,470
 470 GO TO (480,490), NT
 480 PRES=CT*SUMG
 GO TO 530
 490 PRES=CT*SUMGH+CTE*GJ(1)
 C COMPUTATION OF ENERGY TRANSFER TO THE ELECTRON GAS BY ELASTIC
 C COLLISIONS
 TEP=TEP**3
 VIC=VIC1*RHOP/SQRT(TEQ)*ALOG(VIC2*TEQ/RHOP/GJ(1))
 QELAS=0.
 VNCC=VNC*SQRT(TEP)*RHOP*PIOMEG(TEP)
 DO 520 J=2,ISS
 ICHG=ICH(J)
 GO TO (500,510), ICHG
 500 VEJ=VNCC*GJ(J)
 GO TO 520
 510 VEJ=VIC*GJ(J)
 520 QELAS=QELAS+VEJ/CGI(J)
 QELAS=3.*CRA*RHOP*CGI(1)*CTAP*(CT-CTE)*GJ(1)*QELAS

ORIGINAL PAGE IS
 OF POOR QUALITY


```

SUBROUTINE CXSECT (L,J)
SUBROUTINE CXSECT DETERMINES THE CORRESPONDENCE OF SPECIES PAIRS
BETWEEN THE MASTER SPECIES LIST AND THE TRANSPORT CALCULATIONS.
COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)
M=L-J
L=I(L)
J=I(J)
IF (ISIGN(M,L-J).EQ.M) GO TO 10
M=L
J=M
RETURN
END

```

```

CX S
CX S
CX S
CX S
CX S
CX S
CX S
CX S
CX S
CX S
CX S
CX S
CX S

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13-

```

C

10

SUBROUTINE DERIVS
 LOGICAL ERR,FAILED
 REAL ACOM(30),ELMENT(10),HP(20)
 DOUBLE PRECISION AA,CAPX,CDIJ
 COMMON AA(22,24),CDIJ(20,10),CAPX(20)
 COMMON A,AFNLS,AFNX,AMACH,
 C,CARBS,CSTA,CH,CLNT,CLTAP,
 1 CRR,CRS,CTPL,CTT,CX,DLOGR,
 2 CTP,CTPL,CTT,CX,DLOGR,
 3 DELT2,DELTAX,DLOGA,PRESA,
 4 DELT2,DELTAX,DLOGA,PRESA,
 5 PCTEST,PRES,RHOC,RHOP,
 6 RHOC,RHOP,SC,SL,SL64,
 7 SEN,SHPG,SC,SL,SL64,
 8 TEST,TESTB,TPRINT,TSTOP,
 COMMON BE(64),BET(20),BLBK(31),
 1 CCRPJ(20),CEACT(64),CGI(20),
 2 CLN1MC(64),CLNFI(64),CMW(20),
 3 GJ3(20),PERTGJ(20),PGJ(20),
 4 QQ(64),SAJ(20),SDCHI(64),
 5 SKIL(20),SS(20),TFK(20),
 6 TFD(20),TFE(20),TFC(20),
 COMMON BETA(64,20),XNUIJP(64,20)
 1 XNUIJ(64,20),XNUIJP(64,20)
 COMMON IC,INEQ,INEQV,IP,IRUN,
 1 ISMC,ISMCNR,ISR,ISS,ISSNR,
 2 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,
 3 ISW5A,ISW6A,ISW6B,IUPD,IZERO,
 4 NIT,NNS,NQS,NQT,NTEST
 COMMON IGJ(20),IGM(20),ITB(5),
 COMMON ACOM,ELMENT,HP,ITB(5),KUR(64,20),
 COMMON /BL/DELBL(2),BLINT(2),XZERO,
 1 SW,RO,JDIM,IPCINT,TWALL,CPWALL,VISROT,
 COMMON /CERROR/ERR,DIAM(2),
 COMMON /CERROR/ERR,DIAM(2),
 COMMON /AREA/ATPI(11,2),PARAM(3,12,2),
 1 NSECTU(2),ISHAPE(12,2),NPROFL(2),
 COMMON /TNONEQ/CTE,DTE,BPAR,EPAR(2,25),
 1 KTR(25),ICH(20),IPA(25),KTF(25),
 COMMON /TNODC/XMJATD(20,2),CLNTD(2),
 1 CTOD(2),CTD(2)
 DIMENSION DELI(2),DELI2(2),DELO1(2),
 NAMELIST /OKVDMP/ITRY,DELBL,DELI,DELI2,DELO1,DELO2,DEN
 ITRY=0
 IF (NT.EQ.1) GO TO 30
 CTSAVE=CT
 CT=CTE
 CALL THERM
 CLNTD(2)=CLNT
 HJE=SHJ(1)
 CPJE=CCPJ(1)
 SENTE=SENT(1)
 DO 20 I=1,ISS
 XMJATD(I,2)=XMJAT(I)
 CT=CTSAVE
 CALL THERM
 CLNTD(1)=CLNT

40	CTD(1)=CT	DER 56
	DO 40 I=1,ISS	DER 57
	XMJATD(I,1)=XMJAT(I)	DER 58
	IF (NT.EQ.1) GO TO 50	DER 59
	SHJ(1)=HJE	DER 60
	CCPJ(1)=CPJE	DER 61
	SENT(1)=SENTE	DER 62
	CTD(2)=CTE	DER 63
50	CALL COMM	DER 64
	IF (ERR.OR.FAILED) RETURN	DER 65
	CALL EXACT	DER 66
	IF (ERR) RETURN	DER 67
	IF (DLGGA*CX.GE.-0.01) GO TO 60	DER 68
	FAILED=.TRUE.	DER 69
	RETURN	DER 70
60	IF (ISW38.EQ.0.CR.IUPD.EQ.1) GO TO 140	DER 71
	DO 70 K=1,NPRFLS	DER 72
	DEL12(K)=DEL11(K)	DER 73
70	DEL11(K)=DELBL(K)	DER 74
	CALL BLCALL(.FALSE.)	DER 75
	IF (ERR) RETURN	DER 76
	IF (ABS(1.-DELBL(1)/DEL11(1)).GT.0.01) GO TO 80	DER 77
	IF (NPRFLS.EQ.1) GO TO 130	DER 78
	IF (ABS(1.-DELBL(2)/DEL11(2)).LE.0.01) GO TO 130	DER 79
80	DO 90 K=1,NPRFLS	DER 80
	DELO2(K)=DELO1(K)	DER 81
90	DELO1(K)=DELBL(K)	DER 82
	ITRY=ITRY+1	DER 83
	IF (ITRY.EQ.1) GO TO 10	DER 84
	IF (ITRY.GT.6) GO TO 110	DER 85
	DO 100 K=1,NPRFLS	DER 86
	DEN=DELO2(K)-DELO1(K)-DEL12(K)+DEL11(K)	DER 87
	IF (DEN.EQ.0.) GO TO 120	DER 88
100	DELBL(K)=(DEL11(K)*DELO2(K)-DELO1(K)*DEL12(K))/DEN	DER 89
	GO TO 10	DER 90
110	WRITE (6,150)	DER 91
120	*WRITE (6,DRVDMPI)	DER 92
130	CONTINUE	DER 93
140	RETURN	DER 94
C		DER 95
C		DER 96
C		DER 97
150	FORMAT (39H BOUNDARY LAYER ITERATION NOT CONVERGED)	DER 98
	END	DER 99-

```

SUBROUTINE DSMSCL (A,I,N)
DOUBLE PRECISION A
BASED ON SUBROUTINE SIMQ FROM IBM SCIENTIFIC SUBROUTINE PACKAGE
DATA RNAME /6HDSMSCL/
DIMENSION A(22,24)
I=N+1
M1=I+1
L=0
DO 20 K=1,I
DO 10 J=1,I
LN=L/N
L=L+1
LM=L-N*LN
LNI=LN+1
A(LM,LNI)=A(J,K)
IF (K.EQ.1) A(J,N1)=A(J,M1)
CONTINUE
CONTINUE
CALL SIMQ (A(1,1),A(1,N1),I,KS)
IF (KS.EQ.0) GO TO 30
WRITE (6,50)
CALL DUMP (RNAME)
RETURN
DO 40 J=1,I
A(J,M1)=A(J,N1)
RETURN

```

```

FORMAT (36HOMATRIX OF COEFFICIENTS IS SINGULAR.)
END

```

1 DSM
 2 DSM
 3 DSM
 4 DSM
 5 DSM
 6 DSM
 7 DSM
 8 DSM
 9 DSM
 10 DSM
 11 DSM
 12 DSM
 13 DSM
 14 DSM
 15 DSM
 16 DSM
 17 DSM
 18 DSM
 19 DSM
 20 DSM
 21 DSM
 22 DSM
 23 DSM
 24 DSM
 25 DSM
 26 DSM
 27 DSM
 28 DSM
 29 DSM
 30- DSM

SUBROUTINE DUMP (RNAME)
 LOGICAL ERR
 COMMON /ERROR/ ERR
 ERR=TRUE
 WRITE (6,10) RNAME
 RETURN

C
 C
 10

FORMAT (1H1,24H DUMP ROUTINE CALLED BY ,A6)
 END

DUM
 DUM
 DUM
 DUM
 DUM
 DUM
 DUM
 DUM
 DUM
 DUM
 10-

ORIGINAL PAGE IS
 OF POOR QUALITY

```

SUBROUTINE DUMPEX
LOGICAL DATAP, FAILED, COUPLD, NOREAC, MODLPT
DOUBLE PRECISION AA, CAPX, CDIJ, AIN
COMMON AA(22,24), CDIJ(20,10), CAPX(20)
COMMON A, AFNTS, CHA, CLNT, AR, ARBA, ARBB, BZERO,
1 CRRB, CRS, CSTA, CT, CTAP, CM, CMA, CRA, CRP,
2 CTP, CTPL, CTT, CX, CXB, CXMAX, DATEST, DBTEST, CTMX,
3 DELT, DELTAX, DLOGA, DLOGR, PRESB, PRESTH, PRHC, FLUX, HDELX, DELT1,
4 SPTTEST, PRES, PRESA, RHOP, RHPL, RHTH, ROBARA, ROBARP, RHO,
5 RHOB,
6 RHOC, SHPG, SC, SL, SL64, SM, SU, SDT,
7 SEN, TESTB, TPRINT, TSTOP, UP, ZP, ZPA, SUMG,
8 TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA, SUMG,
COMMON RE(64), BET(20), BLBK(31), CAI(64), CAPXTH(20),
1 CCPJ(20), CEACT(64), CGI(20), CGMU(20), CHI(64),
2 CLNIMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJ(20),
3 GJB(20), PERTGJ(20), PGJ(20), PI(64), PICH(64),
4 GQ(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20),
5 SKIL(20), SS(20), TFE(20), TFK(20), THEV(20), TFA(20),
6 TFD(20), TFE(20), TFK(20), THEV(20), TFA(20),
COMMON BETA(64,20), ELJ(10,20), GELJ(10,20),
1 XNUIJ(64,20), XNUIJP(64,20),
COMMON IC, INEQ, INEQV, IP, IRUN, ISC,
1 ISMC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3,
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B,
3 ISW5A, ISW5B, ISW6A, ISW6B, IUPC, IZERO, JJK,
4 NIT, NNN, NNS, NQS, NOT, NTEST, LC, M1, NFIT,
COMMON IGJ(20), IGM(20), ITB(5), KUR(64,20), LPIJ(20,10),
COMMON TEMPRY/SAVEC(20),
COMMON /BL/DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SW, RC, JDIM, IPCINT,
COMMON /TRPROP/ VISC, PR, SIGMA, FLEWIS
COMMON /NEQ/ OMDST(2), DELBL(2),
COMMON /AREA/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12,2), NPROFL(2), NPRELS, NBL
COMMON /EOC/ ZCAP(20), ZSEN, ZCH, ZCM, ZRBP, ZRHO
COMMON /STAG/ TS, EPSLON, TSF, EPSF
COMMON /MODP2/ MODLPT
COMMON /MODPAR/ XMPI, DXMP, FSTAG, CATFAC, TMODEL, XMODP1, DXMODP,
1 TPLATE, KDIM
COMMON /NEWMP/ FACMP, NMDOPT
COMMON /VODPT/ TSOIAM(20), TSAR(20), NTS, MBL
COMMON /CERROR/ FAILED
COMMON /GLIM2/ NCREAC(64)
COMMON /BLNF/ XI(2), ORDINP(2), DELBLP(2), DBBLP(2), AMP, AMP,
1 COUPLD, ISMD
COMMON /DRV/ SHJGGJ
COMMON /BLOUT/ REPF, THETA(2), SN(2), XSN(2), PRREF, HR, /DOT(2),
1 TAUW(2)
COMMON /SWITCH/ XPB, DXPB, W
COMMON /CONVRT/ CF(6)
COMMON /TAPOUT/ XXX(10), ITPOUT, NRCOUT, IFLOW, ITPER, IMP, DATAP
COMMON /AEGECM/ SORTA, S1, S2
COMMON /THRT/ RSA
COMMON /TNONEQ/ CTE, DTE, BPAR, EPAR(2,25), NT, ITR(25), KTF(25),

```

```

1 KTR(25), ICH(20), IPA(25)
COMMON /TNDK/ XMJATD(20,2), CLNTD(2), CTD(2)
COMMON /TNERK/ SOTE, CTEB, OEMAX
COMMON /TNCE/ SUMGH, SCPGH, ODPR, QDPE
COMMON /INGNE/ AIN(10,10)
NAMELIST /DMP1/
1 A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO,
2 C, CARB, CH, CM, CMA, CRA, CRP,
3 CRRB, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMXX,
4 CTP, CTPX, CTT, CX, CXB, CXMAX, DATEST, DBTEST, DELT1,
5 DELT2, DELTAX, DLOGA, DLOGR, PRESTH, DT, ENT, FLUX, HDELX, PCT,
6 PCTEST, PRES, PRESA, RHOP, RHOC, RHOB, RHO,
7 RHOBAR, RHOC, RHOP, RHPL, RHTH, ROBAR, SCPG, SDT,
8 SEN, SHPG, SC, SL, SL64, SM, SU, SUMG,
9 TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA
NAMELIST /DMP2/
1 BE, BET, BLBK, CAI, CAPX, CAPXTH, CCPJ,
2 CEACT, CGI, CGMU, CHI, CHII, CLNIMC, CLNFI, C'W, ETAI, ETAJ, GJA,
3 GJB, PERTGJ, PGJ, PI, PICH, QM, QQ, SAJ, SCCHI, SENT, SHJ,
4 SHJA, SKIL, SS, TB, TFA, TFB, TFC, TFD, TFE, TFK, THEV, XMJAT, XNUI
NAMELIST /DMP3/
1 IC, INEQ, INEQV, IP, IRUN, ISC, ISCP1,
2 ISMC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
3 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
4 ISW5B, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, MI, NFIT,
5 NIT, NNN, NNS, NOS, NOT, NTEST, IGJ, IGM, ITB
NAMELIST /DMP4/ DELBL, BLINT, TWALL, CPWALL, VISROT, RO,
1 DIAM, SW, JDIM, IPOINT, VISC, SAVEC, XZERO, PRREF, HR,
2 DMST, DDELBL, ZCAP, ZSEN, ZCH, ZCM, ZRBP, ZRHO, TS, EPSLON,
3 TSF, EPSF, THETA, SN, XSN, REPF, QWDCI, TAUW, SHJ, DGJ,
4 XRB, DXPR, CF, XMP1, FSTAG, CATFAC, MODEL, XMODPI,
5 KDIM, ATIP, PARAM, RTHCN, NSECT, ISHAPE, PR, SIGMA,
6 FLEXIS, FACMP, NMDDPT, TSDIAM, NTS, TSAR, NPROFL, NPRELS, NBL, SORTA,
7 S1, S2, RSA, W, SUMGH, SCPGH, ODPR, ODPE, SCTE, CTEB, CTE, DTE, BPAR,
8 EPAR, NT, ITR, KTF, KTR, ICH, IPA, XMJATD, CLNTD, CTD, OEMAX, AIN,
9 FAILED, NOREAC, XI, ORDINP, DELBLP, ODBLP, AMPP, AMP, COUPLD, ISMD
WRITE (6, DMP1)
WRITE (6, DMP2)
WRITE (6, DMP3)
WRITE (6, DMP4)
IF (.NOT. DATAPE) GO TO 20
DO 10 I=1, NRCCUT
BACKSPACE ITPOUT
WRITE (6, 30) ITPOUT, NRCCUT
NRCCUT=0
WRITE (6, 40)
RETURN
10
20
C
C
C
30
40
END

```

```

FORMAT (11H0*****TAPE,13,14H BACKSPACED BY,14,41H RECORDS TO DELETED
1TE DATA FROM FAILED CASE.)
FORMAT (1H1)
END

```

1 2 3 4 5 6 7 8 9 10 11 12-
 ELC ELC ELC ELC ELC ELC ELC ELC ELC ELC ELC ELC

SUBROUTINE ELCOND (SIGMA,SSIG)
 SUBROUTINE ELCOND COMPUTES ELECTRICAL CONDUCTIVITY IN MHOS/CM.
 COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
 COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)
 COMMON /TRANS5/ N,IELEC,IDI,ID2
 SIGMA=0.
 IF (IELEC.EQ.0) GO TO 20
 DO 10 I=2,N
 SIGMA=SIGMA+X(I)*Q(1,1,I)
 SIGMA=SSIG*X(1)/SIGMA
 RETURN
 END

C 10 20

ORIGINAL PAGE IS
 OF POOR QUALITY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31-
ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	ELT	


```

SUBROUTINE E  VE (ET,IP)
TIMING ROUTE.

ET=ELAPSED TIME FROM BEGINNING OF EXECUTION IN SECONDS
IP=CONTROL FOR PRINTING THE TIME
  IP = 0      SUPPRESS PRINT

  LOGICAL CALLED
  DATA CALLED ,.FALSE./
  ET=ETO
  IF (CALLED) GO TO 10
  ET=0.
  CALL RESET
  CALLED=TRUE.
  ET=0.
  GO TO 30
  CALL TIME (I)
  ET=C.001*I
  IF (IP.EQ.0) GO TO 20
  ET=ET/60.
  DET=ET-ETO
  WRITE (6,40) ETM,DET.
  GO TO 30
  ETO=ETP
  RETURN

10
20
30
40
  
```



```

FORMAT (24H0*****ELAPSED TIME=,F6.2,28H MINUTES SINCE START
1F RUN.,F5.0,32H SECONDS SINCE LAST PRINTED TIME/)
END
  
```

[illegible]

IR = INDI OF REACTIONS
TE = ELEC RUN TEMPERATURE (DEG. K)

OUTPUTS

	ENERGY GAINED BY THE ELECTRON GAS IN THE FORWARD REACTION (CAL/MOLE)	ENERGY GAINED BY THE ELECTRON GAS IN THE REVERSE REACTION (CAL/MOLE)	ENERGY LOST BY RADIATION IN THE FORWARD REACTION (MOLE)	ENERGY LOST BY RADIATION IN THE REVERSE REACTION (MOLE)
E =				
-ER =				
QF =				
-QR =				

```
DATA HR/2.9808/, RO /1.9872/  
IT=IIR{IR}  
IEO=EPAR(1,IR)  
EGO TO (10.20.50.20.60.90). IT  
EFFF=-EPAR(?,IR)*RO*TE
```

```

IF (IT.EQ.4) GO TO 70
DEF=-THR*TF

```

•
O
=
R
W

33-00000

●
○
○
○
○

GO TO 100

FILE

●
○
H
K
W

● ○ ●
○ =
L O

GO TO 40

U
W
Y
X
W

69-100-01

2000

2000

20
 21
 22
 23
 24

● ● ●
○ ○ ○
■ ■ ■
□ □ □

40

NORTH

W
Z
O

3

ORIGINAL PAGE IS
OF POOR QUALITY


```

SUBROUTINE EQCALC (T,P)
LOGICAL ERR
CALCULATES THERMOCHEMICAL EQUILIBRIUM FOR SPECIFIED T AND P.
T=TEMPERATURE IN DEG. K
P=PRESSURE IN ATM.
REAL ACCM(30),ELEMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFNX,AMACH,
C,CARB,CH,CHA,CLNT,
1 CRRB,CRS,CSTA,CT,CTAP,AR,ARBA,ARBB,BZERO,
2 CTP,CTPL,CTT,DLOGA,CX,CTAR,CM,CMA,CRA,CRP,
3 DELT2,DELTA,DLOGA,DLOGR,DT,ENT,FLUX,HDELX,DELT1,
4 SPCTEST,PRESA,PRESB,PRESH,PRMC,RHAP,RHO,RHOB,
5 RHOBAR,RHOC,RHOP,RHPL,RHTR,ROBAR,ROARP,SCPG,SOT,
6 SEN,SHPG,SC,SL,SL64,SU,SU2,SUMG,
7 TEST,TESTB,TPRINT,TSTOP,UP,
8 TEST,TESTB,TPRINT,TSTOP,UP,
COMMON BE(64),BET(20),BLBK(31),CAI(64),CAPXTH(20),ECC
1 CCPJ(20),CEACT(64),CGI(20),CGMU(20),CHI(64),CHI(20),ECC
2 CLNIMC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),ECC
3 GJ(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),ECC
4 QQ(64),SAJ(20),SDCHI(64),SENT(20),SHJ(20),SHJA(20),ECC
5 SKIL(20),SS(20),TFK(20),THEV(20),TFB(20),TFC(20),ECC
6 TFD(20),TFE(20),THEV(20),XNJAT(20),XNU(64),ECC
COMMON BETA(64,20),EIJ(10,20),GELJ(10,20),ECC
1 XNUIJ(64,20),XNUJJP(64,20),ECC
COMMON IC,INEQ,INEQV,IP,IRUN,ISC,ISCP1,
1 ISMC,ISMCNR,ISR,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,
2 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,ISW5A,
3 ISW5B,ISW6A,ISW6B,IUPD,IZERO,IJK,LC,M1,NFI,
4 NIT,NNN,NNS,NOS,NQT,NTEST
COMMON IGM(20),ITB(5),KUR(64,20),LPIJ(20,10),ECC
COMMON ACCM,ELEMENT,HP
DIMENSION AA(22,24),BTA(64,20),CAPO(31),CCI(20),DGJ(20),SRECC
1 J(20),SDGJ(20),SHJAP(20),THEVP(20),ECC
EQUIVALENCE (AA(1,1),AAA(1,1)),(BTA(1,1),BT(1,1)),
1 (BLBK(1),CAPO(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SHJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
COMMON /ERROR/ ERR
COMMON /EQC/ ZCAP(20),ZSEN,ZCH,ZCM,ZRBP,ZRHO
COMMON /EQC2/ ZPZ,ZGMU(20)
DATA RNAME /6HEQCALC /
CTSAVE=CT
CT=T/CTAP
DO 10 I=1,ISC
2 ZCAP(I)=QM(I)
IF (QM(I).LE.0.) ZCAP(I)=1.E-3
ZGMU(I)=ALOG(ZCAP(I))
ZPZ=ALOG(P)
CALL THERM
IF (ISWC.EQ.0) GO TO 60
DO 30 I=1,ISMIC
IL=I+ISC
CHI(I)=-XNJAT(IL)+ZPZ*BET(I)

```

```

21 DO 20 J=1,ISC
30 CHII(I)=CHII(I)+XMJAT(J)*CDIJ(I,J)
CONTINUE
DO 50 I=1,ISMC
SKIL(I)=CHII(I)
DO 40 J=1,ISC
SKIL(I)=SKIL(I)+CDIJ(I,J)*ZGMU(J)
L=I+ISC
ZGMU(L)=SKIL(I)
SKIL(I)=EXP(SKIL(I))
ZCAP(L)=SKIL(I)
M1=ISC+1
N=C
N=N+1
DO 90 J=1,ISC
AA(J,M1)=QM(J)-ZCAP(J)
IF (ISMC.EQ.0) GO TO 90
DO 80 I=1,ISMC
AA(J,M1)=-(CDIJ(I,J)-QM(J)*BET(I))*SKIL(I)+AA(J,M1)
CONTINUE
DO 140 J=1,ISC
DO 140 K=1,ISC
IF (J-K) 110,100,110
AA(J,K)=ZCAP(J)
GO TO 120
AA(J,K)=0.0
IF (ISVC.EQ.0) GO TO 140
DO 130 I=1,ISMC
AA(J,K)=AA(J,K)+(CDIJ(I,K)*(CDIJ(I,J)-QM(J)*BET(I))*SKIL(I))
CONTINUE
CALL DSM SOL (AA,ISC,22)
IF (ERR) RETURN
DO 180 K=1,ISC
ZA=1.+AA(K,M1)
IF (ZA) 150,150,160
ZCAP(K)=ZCAP(K)/2.
GO TO 170
ZCAP(K)=ZCAP(K)*ZA
ZGMU(K)=ALOG(ZCAP(K))
CONTINUE
IF (ISMC.EQ.0) GO TO 210
DO 200 I=1,ISMC
SKIL(I)=CHII(I)
DO 190 L=1,ISC
SKIL(I)=SKIL(I)+CDIJ(I,L)*ZGMU(L)
J=I+ISC
ZGMU(J)=SKIL(I)
SKIL(I)=EXP(SKIL(I))
ZCAP(J)=SKIL(I)
CONTINUE
DO 220 K=1,ISC
IF (ABS(AA(K,M1)).LE.TEST) GO TO 220
IF (N-NTEST) 70,70,240
CONTINUE
ZCH=0.

```

```

E0C 56
E0C 57
E0C 58
E0C 59
E0C 60
E0C 61
E0C 62
E0C 63
E0C 64
E0C 65
E0C 66
E0C 67
E0C 68
E0C 69
E0C 70
E0C 71
E0C 72
E0C 73
E0C 74
E0C 75
E0C 76
E0C 77
E0C 78
E0C 79
E0C 80
E0C 81
E0C 82
E0C 83
E0C 84
E0C 85
E0C 86
E0C 87
E0C 88
E0C 89
E0C 90
E0C 91
E0C 92
E0C 93
E0C 94
E0C 95
E0C 96
E0C 97
E0C 98
E0C 99
E0C 100
E0C 101
E0C 102
E0C 103
E0C 104
E0C 105
E0C 106
E0C 107
E0C 108
E0C 109
E0C 110

```

EOC 111
 EOC 112
 EOC 113
 EOC 114
 EOC 115
 EOC 116
 EOC 117
 EOC 118
 EOC 119
 EOC 120
 EOC 121
 EOC 122
 EOC 123
 EOC 124
 EOC 125
 EOC 126
 EOC 127
 EOC 128
 EOC 129-

```

ZSEN=-ZPZ
ZCM=0.
DO 230 I=1,ISS
  ZSEN=ZSEN+ZCAP(I)*(SENT(I)-ZGMU(I))
  ZCH=ZCH+ZCAP(I)*SHJ(I)
  ZCM=ZCM+ZCAP(I)*CGI(I)
  ZRHP=P*ZCM/T/R2.055
  ZRHO=ZRBP/(1.+ZRBP*BZERO)
  ZCH=ZCH+(ZRBP/ZRHO-1.)
  CT=CT+SAVE
  RETURN
WRITE (6,250)
CALL DUMP (RNAME)
RETURN
      FORMAT (35H100 MANY NEWTON-RAPHSON ITERATIONS)
      END
  
```

230

240

C C

C 250

ORIGINAL PAGE IS
 OF POOR QUALITY

```

SUBROUTINE EXACT
LOGICAL ERR
REAL ACOM(30),ELMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFNX,AMACH,AR,ARBA,ARRB,BZERO,
1 C,CRRB,CRS,CSTA,CH,CHA,CLNT,CM,CTB,CTC,CNA,CRA,CRP,
2 CTP,CTPL,CTT,CT,CTAP,CXB,CXMAX,DATEST,OBTEST,DELT1,
3 DELT2,DELTAX,DLOGA,DLOGR,DT,ENT,FLUX,HDELX,PCT,
4 SPCTEST,PRES,PRESA,PRESTH,PRHO,RHAP,RHO,
5 RHOBAR,RHOC,RHOP,RHPL,RHTH,ROBAR,ROBAP,SCPG,SDT,
6 SEN,SHPG,SC,SL,SL64,SU,SU2,SUMG,
7 TEST,TESTB,TPRINT,TSTOP,ZP,ZPA
8 COMMON BE(64),BLBK(31),CAI(64),CAPXTH(20),EXA
1 CCPJ(20),CEACT(64),CGI(20),CGMU(20),CHI(64),CHI(20),EXA
2 CLNIMC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),EXA
3 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),OM(20),EXA
4 QQ(64),SAJ(20),SDCH(64),SENT(20),SHJ(20),SHJA(20),EXA
5 SKIL(20),SS(20),TB(30),TFA(20),TFB(20),TFEC(20),EXA
6 TFD(20),TFE(20),TFK(20),THEV(20),XNJAT(20),XNUL(64)
7 COMMON HETA(64,20),ELJ(10,20),GELJ(10,20),EXA
1 XNUIJ(64,20),XNUIJP(64,20),INEQ,INEQV,IP,IRUN,ISC,ISCP1,
2 COMMON IC,IM,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,EXA
1 ISMC,JSMCNR,ISR,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,EXA
2 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,ISW5A,EXA
3 ISW5B,ISW6A,ISW6B,IUPD,IZERD,JJK,LC,M1,NFIT,EXA
4 NIT,NNN,NNNS,NQC,NOT,NTEST,KUR(64,20),LPIJ(20,10)
5 COMMON IGJ(20),IGM(20),ITB(5),KUR(64,20),LPIJ(20,10)
6 COMMON ACOM,ELMENT,HP
7 DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),SBE
1 J(20),SDGJ(20),SHJAP(20),THEVP(20)
2 EQUIVALENCE (AA(1,1),AAA(1,1)),(HETA(1,1),BTA(1,1)),
3 (BLRK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
4 (CAPX(1),GJ(1)),(SAJ(1),SRJ(1)),(SS(1),SDGJ(1)),
5 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
6 COMMON /DRV/ SHJDGJ
7 COMMON /ERROR/ ERR
8 COMMON /SS/ CAS,US
9 COMMON /TNERK/ SDTE,CTEB,DCHA,CHB,SDCHA,DQMAX,IFAIL
10 COMMON /TNOREQ/ CTE,DTE,BPAR,EPAR(2,25),NT,ITR(25),KTF(25),
11 KTR(25),ICH(20),IPA(25)
12 COMMON /TNCE/ SUMGH,SCPGH,GDPR,ODPE
13 NAMELIST /EXTDMP/ CT,CMA,SU2,DLOGA,DLOGR,SUMG,SHJDGJ,SUMTDG,
14 IUPD,NT,DT,DTE,DGJ,PICHI,SHJ
15 Z1=SU2/CMA
16 IF (INEQ.NF.0) GO TO 100
17 DO 10 I=1,ISSP2
18 DO 10 J=1,ISSP4
19 DO 20 I=1,ISSC
20 DO 20 J=1,ISS
21 AAA(I,J)=LPIJ(J,I)
22 DO 50 I=ISSP1,ISS
23 K=I-ISC

```

EXA 56
EXA 57
EXA 58
EXA 59
EXA 60
EXA 61
EXA 62
EXA 63
EXA 64
EXA 65
EXA 66
EXA 67
EXA 68
EXA 69
EXA 70
EXA 71
EXA 72
EXA 73
EXA 74
EXA 75
EXA 76
EXA 77
EXA 78
EXA 79
EXA 80
EXA 81
EXA 82
EXA 83
EXA 84
EXA 85
EXA 86
EXA 87
EXA 88
EXA 89
EXA 90
EXA 91
EXA 92
EXA 93
EXA 94
EXA 95
EXA 96
EXA 97
EXA 98
EXA 99
EXA 100
EXA 101
EXA 102
EXA 103
EXA 104
EXA 105
EXA 106
EXA 107
EXA 108
EXA 109
EXA 110

```

30  S1=BET(K)+SHJ(I)/CT
    DO 30 J=1,ISS
      AAA(I,J)=CDIJ(K,J)*GJ(J)
    S1=S1-CDIJ(K,J)*SHJ(J)/CT
    DO 50 J=ISSP1,ISS
      IF (I-J) 50,40,50
    AA(I,J)=-1.0/GJ(J)
    CONTINUE
    AA(I,ISSP1)=S1/CT
    AA(I,ISSP2)=RET(K)
    CONTINUE
    DO 70 J=1,ISS
      AAA(ISSP1,J)=1.0
      AAA(ISSP2,J)=SHJ(J)
      AAA(ISSP1,ISSP1)=SUMG/CT
      AAA(ISSP1,ISSP2)=SUMG-Z1/CT
      AAA(ISSP1,ISSP3)=-Z1/CT
      AAA(ISSP2,ISSP1)=SCPG
      AAA(ISSP2,ISSP2)=-Z1
      AAA(ISSP2,ISSP3)=-Z1
    DO 80 I=ISSP1,ISSP2
      AAA(I,ISSP3)=-DLOGA*AAA(I,ISSP3)
      CALL DMSOL (AAA,ISSP2,22)
      IF (ERR) RETURN
    DO 90 I=1,ISS
      DGJ(I)=AAA(I,ISSP3)
      DT=AAA(ISSP1,ISSP3)
      DLOGR=AAA(ISSP2,ISSP3)
      GO TO 180
    NE=NT+1
    IF (NT.EQ.2) DCHA=-(CMA/CRP/RHAP/US)*QDPR/RHO/SU
    SHJDGJ=0.
    SUMTDG=0.
    DO 120 J=1,ISS
      DGJ(J)=0.
    DO 110 I=1,ISR
      DGJ(J)=DGJ(J)+BETA(I,J)*PICH(I)
    SHJDGJ=SHJDGJ+SHJ(J)*DGJ(J)
    IF (J.LT.NT) GO TO 120
    SUMTDG=SUMTDG+DGJ(J)
    CONTINUE
    SUMTDG=CT*SUMTDG
    IF (NT.EQ.2) SUMTDG=SUMTDG+CT*DGJ(1)
    ECOEF=SQRT(CMA/8.31434E7/CTAP)/RHOP/SU/CRP
    AAA(1,1)=SCPGH
    IF (NT.EQ.2) AAA(1,3)=CCPJ(1)*GJ(1)
    AAA(1,2)=-Z1
    IF (IUPD.EQ.1) GO TO 130
    AAA(1,NE+1)=Z1*DLOGA-SHJDGJ
    IF (NT.EQ.1) GO TO 130
    AAA(1,4)=AAA(1,4)-ECOEF*QDPR
    AAA(2,1)=SUMGH
    IF (NT.FG.2) AAA(2,3)=GJ(1)
    IF (IUPD.EQ.1) GO TO 140
    AAA(2,2)=CT*SUMGH-Z1+(NT-1)*GJ(1)*CTE

```

EXA 111
EXA 112
EXA 113
EXA 114
EXA 115
EXA 116
EXA 117
EXA 118
EXA 119
EXA 120
EXA 121
EXA 122
EXA 123
EXA 124
EXA 125
EXA 126
EXA 127
EXA 128
EXA 129
EXA 130
EXA 131
EXA 132
EXA 133
EXA 134
EXA 135
EXA 136
EXA 137
EXA 138
EXA 139-

```

140  AAA(2,NE+1)=-SUMTDG+Z1*DLOGA
      IF (NT.EQ.1) GO TO 150
      AAA(3,1)=0.
      AAA(3,3)=1.5*GJ(1)
      IF (IUPD.EQ.1) GO TO 150
      AAA(3,2)=-GJ(1)*CTE
      AAA(3,4)=-1.5*CTE*DGJ(1)+ECOE*QDPE
      IF (IUPD.EQ.0) GO TO 170
      AAA(1,NE+1)=Z1*DLOGR-SHJDGJ
      IF (NT.EQ.1) GO TO 160
      AAA(1,4)=AAA(1,4)-ECOE*QDPR
      AAA(2,2)=-Z1
      AAA(2,NE+1)=-SUMTDG-DLOGR*(CT*SUMGH-Z1+(NT-1)*GJ(1)*CTE)
      AAA(3,2)=0.
      IF (NT.EQ.1) GO TO 170
      AAA(3,4)=GJ(1)*CTE*DLOGR-1.5*CTE*DGJ(1)+ECOE*QDPE
      CALL DSM SOL (AAA,NE,22)
      IF (ERR) RETURN
      DT=AAA(1,NE+1)
      IF (IUPD.EQ.0) DLOGR=AAA(2,NE+1)
      IF (IUPD.EQ.1) DLOGA=AAA(2,NE+1)
      IF (NT.EQ.2) DTE=AAA(3,NE+1)
      IF (DLOGA.NE.0.) GO TO 190
      AMACH=1.
      GO TO 200
      AMACH=1./SQRT(ABS(1.+DLOGA/DLOGR))
      IF (INEQ.NE.0.AND.ISW5B.NE.0) WRITE (6,EXTDMP)
      RETURN
      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

1 2 3 4 5 6 7 8-
EXP
EXP
EXP
EXP
EXP
EXP
EXP

FUNCTION EXP(X)
IMPLICIT REAL*8 (A-H,O-Z)
IF (X.GT.-180.) GO TO 10
EXP=0.D0
GO TO 20
EXP=DEXP(X)
RETURN
END

10
20

ORIGINAL PAGE IS
OF POOR QUALITY

```

C C C C C C C
SUBROUTINE FINDX (A,UPDOWN,X)
LOGICAL ERR,UPPER
SOLVES GEOMETRIC AREA RELATION FOR X, GIVEN THE VALUE A(X).
A=GEOMETRIC AREA RATIO (=1.0 AT GEOMETRIC THROAT)
UPDOWN=1. IF DOWNSTREAM SOLUTION IS DESIRED
=-1. IF UPSTREAM SOLUTION IS DESIRED
X=VALUE OF X OBTAINED
ENTRY FINCXC SOLVES FOR X AT WHICH THE MBL-TH PROFILE HAS A
HALF-WIDTH OF A.
COMMON /ERROR/ ERR
DIMENSION Z(2),DZDX(2),RNAME(2)
DATA RNAME /SHFINDX,6HFINDXC/
NAMELIST /DMP/ V,AR,DADX,VO,VL,VU,ERRL,ERRU,ERR,UPPER
IENTRY=1
GO TO 10
ENTRY FINDXC(A,MBL,X)
IENTRY=2
UPDOWN=1.
N=0
V=100.
GO TO 60
N=0
V=1.
IF (ABS(A-1.0).LT.1.E-5) GO TO 40
IF (A-1.) 20,40,50
WRITE (6,170) A
CALL DUMP (RNAME(IENTRY))
X=0.
RETURN
VL=0.
ERRL=1.-A
UPPER=.FALSE.
ERRU=1.E30
NEWTON-RAPHSON ITERATION TO SOLVE FOR X
N=N+1
IF (N.LE.50) GO TO 70
WRITE (6,180) A,UPDOWN,IENTRY,MBL
WRITE (6,DMP)
GO TO 30
GO TO (80,130), IENTRY
CALL GEOMAR (SIGN(V,UPDOWN),AR,DADX)
IF (ERR) RETURN
ERA=AR-A
IF (ERA) 90,160,110
IF (ERA-ERRL) 140,140,100
ERRL=ERA
VL=V
GO TO 140
UPPER=.TRUE.
IF (ERA-ERRU) 120,140,140
ERRU=ERA
VU=V
GO TO 140
CALL GMAR3 (V,DZDX(1),DZDX(2),Z(1),Z(2))
AR=Z(MBL)

```


140	DADX=DZDX(MBL)	FDX	56
	IF (ABS(1.-AR/A).LE.1.E-5) GO TO 160	FDX	57
	IF (N.LE.10) GO TO 150	FDX	58
	IF (IENTRY.EQ.2.OR..NOT.UPPER) GO TO 150	FDX	59
	V=VL-(VU-VL)*ERHL/(ERRU-ERRL)	FDX	60
	GO TO 60	FDX	61
150	VO=V	FDX	62
	V=V+(A-AR)/ABS(CADX)	FDX	63
	V=AMIN1(V.2.*VO)	FDX	64
	GO TO 60	FDX	65
160	X=SIGN(V,UPDOWN)	FDX	66
	RETURN	FDX	67
C		FDX	68
170	FORMAT (49HCFINDX CALLED WITH AN AREA RATIO LESS THAN UNITY,.10X,2	FDX	69
	1HA=.1PE15.8)	FDX	70
180	FORMAT (33H0MOPE THAN 50 ITERATIONS IN FINDX.10X,2HA=.1PD15.8.10X,FDX	FDX	71
	17HUPDOWN=.0PF3.0.10X,7HENTRY=.12.10X,4HMBL=.12)	FDX	72
	END	FDX	73-

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE FROZEO
REAL ACOM(30),ELMENT(10),HP(20)
LOGICAL SWITCH,SKIP,MODLPT,ISTEPC,AXISYM,WEDGEM,ERR,LASTPT
DOUBLE PRECISION AA,AAA,CAPX,GJ,COIJ
COMMON AA(22,24),COIJ(20,10),CAPX(20)
COMMON A,AFNLS,AFNX,AMACH,AR,ARBA,ARBB,BZERO,
1  CRRG,CRS,CSTA,CTI,CTAP,CM,CMA,CRA,CRP,
2  CTP,CTPL,CTT,CX,CX8,CXMAX,DATFST,DBTEST,DELT,
3  DELT2,DELTAX,DLOGA,DLOGR,DLOGT,ENT,FLUX,HDELX,PCT,
4  SPCTEST,PRES,PRESA,PRESTH,PRHQ,RHAP,RHQ,RHOB,
5  GRHOBAR,RHOC,RHCP,RHPL,RHTH,ROBARA,ROBARP,SCPG,SDT,
6  SEN,SHPG,SC,SL,SL64,SM,SU,SU2,SUMG,
7  TEST,TESTB,TPRINT,TSTOP,UP,ZP,
8  COMMON BE(64),RET(20),BLBK(31),CAI(64),CAPXTH(20),
9  CCPJ(20),CEACT(64),CGI(20),CGMU(20),CHI(64),CHI(20),FRO
10  CLNMC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),FRO
11  GJ(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),FRO
12  QG(64),SAJ(20),SDCHI(64),SENT(20),SHJ(20),FRO
13  SKIL(20),SS(20),TB(30),TFA(20),TFB(20),FRO
14  TFD(20),TFE(20),TFK(20),THEV(20),XMJAT(20),XNUI(64)
15  COMMON BETA(64,20),ELJ(10,20),GELJ(10,20),FRO
16  XNUIJ(64,20),XNUIJP(64,20),FRO
17  COMMON IC,IM,INEQ,INEOV,IP,IRUN,ISC,ISCP1,
18  ISXC,ISMCNR,ISR,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,
19  ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3R,ISW4A,ISW4B,ISW5A,
20  ISW5B,ISW6A,ISW6B,IUPD,IZERO,JUK,LC,MI,NFIT,
21  NIT,NNN,NNS,NQS,NOT,NTEST,KUR(64,20),LPIJ(20,10)
22  COMMON IGM(20),ITB(5),FRO
23  COMMON ACOM,ELMENT,HP,FRO
24  DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),SBFRO
25  IJ(20),SGGJ(20),SHJAP(20),THEVP(20),FRO
26  EQUIVALENCE (AAA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),FRO
27  (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),FRO
28  (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),FRO
29  (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1)),FRO
30  COMMON /TEMPRY/SAVEC(20),FRO
31  COMMON /AREA/ATPI(1,2),PARAM(3,12,2),RTHCM(2),NSECT(2),FRO
32  NSECTU(2),ISHAPE(12,2),NPROFL(2),NPFELS,NBL,FRO
33  COMMON /BLQUT/REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOOT(2),FRO
34  TAUA(2),FRO
35  COMMON /RESPRP/ VISC,PRR,SIGR,FLEWR,FRO
36  COMMON /TRPRRP/ VISC,PR,SIGMA,FLEWR,FRO
37  COMMON /ERROR/ ERH,FRO
38  DIMENSION DELBLO(2),DELBLP(2),FRO
39  COMMON /OUTPUT/ FVOUT(35),GJMF(20),FRO
40  COMMON /MODPAR/ XMP1,DXMP,FSTAG,CATFAC,TMODEL,XMODP1,DXWODP,FRO
41  TPLATE,KDIN,FRO
42  COMMON /STAG/ T2E,EPISLON,T2F,EPSE,FRO
43  COMMON /MCDPT/ TSDIAM(20),TSAR(20),NTS,MBL,FRO
44  DIMENSION BLVAR(13),BLVARC(13),BLVARS(13),FRO
45  EQUIVALENCE (BLVAR(1),REPF),FRO
46  DIMENSION SOLN(3,2),FRO
47  DATA SOLN /4HFROZ,4HEN F,3HLOW,4HEQUI,4HL1BR,3HIUM/,FRO
48  DATA ONE /1./,FRO

```

```

COMMON /BL/ DELBL(2),BLINT(2),XZERO, WALL,CPWALL,VISROT,DIAM(2),
1 SW,RO,JDIM,IPCINT
COMMON /SS/ CAS,US
COMMON /NEQ/ CMDST(2),DDELBL(2)
COMMON /SWTCH/ XPB,DXPB,W
COMMON /AVG/ WSAVE
COMMON /RDWEDG/ ANGLE(10),PADLE(5),WX1,DWX,WX1(20),TWEDGE,WK,
1 NAX,NANGLE,NRAOLE,WEDGEM,AXISYM,ISW9B
1 NAMELIST /FRDMP/ CT,ISONIC,FLAG,PRES,RHO,SU,AMACH,AFNX,XLAST,
1 DELBL,DELRLO,XO,DELBLP,XP,BLVAR,IRSTRT,W
1 NAMELIST /FRDVP2/ XMODEL,X1,X2,CT,CT1,CT2,NMPI,XFRAC,DELBL,
1 DELBLO,ERRX
C FROZEN FLOW SOLUTION
ENTRY FROZEN
ISOLN=1
C DETERMINE SONIC CONDITIONS IN FROZEN FLOW
D-LET=DELT1
F1=0.0
T1=1.0
CT=T1-DELT
S1=C.0
CLNMA=ALOG(CMA)
DO 10 J=1,ISS
SS(J)=ALOG(GJA(J))
S1=S1-GJA(J)*(SS(J)+CLNMA)
S2=-(ZPA/CMA)
ENT=S1+S2-CRRB
T2=CT
CALL THERM
CALL PROP
F2=FLUX
CT=CT-DELT
T3=CT
CALL THERM
CALL PROP
F3=FLUX
IF (F3-F2) 40,60,80
40 IF (DELT-TESTB) 90,50,50
50 DELT=DELT/2.0
CT=T1-DELT
GO TO 20
60 IF (DELT-TESTB) 90,70,70
70 T1=T2
F1=F2
DELT=DELT/2.0
CT=T1-DELT
GO TO 20
80 T1=T2
T2=T3
F1=F2
F2=F3
GO TO 30
90 Z=F1-F2+F3-F2
IF (ABS(Z/F2).GT..00001) GO TO 100
CT=T2

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

100      GO TO 110
110      CT=I2+DELT*(F3-F1)/(2.0*Z)
      CONTINUE
      CALL THERM
      CALL PRCP
      SM=FLUX
      CTMAX=CT
      PRESTH=PRES
      RHTH=PHC
      IPASS=2
      GO TO 120
      EQUILIBRIUM FLOW SOLUTION
      ENTRY EQUIL(IPASS)
      ISCLN=2
      IIRST=1
      W=SAVE
      DO 140 J=1,ISS
      GJ(J)=GJA(J)
      ISCNIC=-1
      LASTPT=.FALSE.
      SKIP=.FALSE.
      DELTSV=DELT1
      IF (ISW3H.NE.0) DELT1=0.049*(1.-CTMAX)
      XMCD1=XMPI
      ITS=1
      ISTEPC=.FALSE.
      MODLPT=.FALSE.
      CALL NEXTMP (ITS,XMCD1,XMODEL)
      INEQ=C
      T2E=CTAP
      DO 150 I=1,13
      BLVAR(I)=0.
      EPSLON=C.
      T2F=CTAP
      EPSFEU.
      DO 160 I=1,NPRFLS
      CCLBL(I)=0.
      OMDS(I)=1.
      ISWICH=0
      SWTCH=.FALSE.
      IPOINT=1
      PRES=1.0
      CT=1.0
      RHU=1.0
      RHODAP=ROBARA
      SU=0.0
      AVACH=0.0
      AFNX=C.0
      RHOS=1.0
      PRESB=1.0
      DELTV=DELT1
      XO=0.
      XP=0.
      FLAG=-1.
      IF (IPASS.EQ.1) GO TO 180
      FROM 111
      FROM 112
      FROM 113
      FROM 114
      FROM 115
      FROM 116
      FROM 117
      FROM 118
      FROM 119
      FROM 120
      FROM 121
      FROM 122
      FROM 123
      FROM 124
      FROM 125
      FROM 126
      FROM 127
      FROM 128
      FROM 129
      FROM 130
      FROM 131
      FROM 132
      FROM 133
      FROM 134
      FROM 135
      FROM 136
      FROM 137
      FROM 138
      FROM 139
      FROM 140
      FROM 141
      FROM 142
      FROM 143
      FROM 144
      FROM 145
      FROM 146
      FROM 147
      FROM 148
      FROM 149
      FROM 150
      FROM 151
      FROM 152
      FROM 153
      FROM 154
      FROM 155
      FROM 156
      FROM 157
      FROM 158
      FROM 159
      FROM 160
      FROM 161
      FROM 162
      FROM 163
      FROM 164
      FROM 165

```

```

170 WRITE (6,630) (ISOLN(N,ISCLN),N=1,3)
DO 170 J=1,ISS
SAVEC(J)=GJA(J)*CMA
AMACH=0.
AFNX=1.F38
VICC=VISC
SIGMA=SIGR
CX=-1.E10
CH=CHA
CM=CMA
CALL OUT2 ('ISCLN')
IF (ERR) RETURN
DECREMENT TEMPERATURE TO GENERATE SOLUTION
IF (.NOT. TSTEPC.AND.(XO-XP).LT.10.) GO TO 190
TSTEPC=.TRUE.
DELVE=AMINI((DELT1,1C.*DELTV/(XO-XP)))
DELVE=AMAXI((DELT1,C.02*DELT1))
DELVE=AMINI((DELT1,0.1*CT))
CT=CT-DELTV
IF (ISONIC.EQ.-1.AND.CT.LE.CTMAX) GO TO 200
IF (CT.LT.TSTOP) GO TO 610
GO TO 220
THROAT HAS BEEN REACHED. COMPUTE CONDITIONS AT THROAT.
IF (ISW3B.NF.0) GO TO 210
FLAG=1.
ISONIC=1
GO TO 220
CT=STARECT+DELTV
CT=CTMAX
ISONIC=0
COMPUTE THE FLOW CONDITIONS FOR TEMPERATURE CT.
GO TO (230,240), ISCLN
CALL THERM
CALL PROC
GO TO 250
CALL NEWRAP
IF (ERR) RETURN
AFNX=AVAXI(SM/FLUX,ONE)
IF (PRES.EQ.PRES) GO TO 260
AMACH=SU*SQRT(RHO*ABS(ALCG(RHO/RHOB)/ALCG(PRES/PRESB)))/PRES)
PRES=PRES
RHO=RHOB
ASE=SU*US/AMACH
CASE=AS/CT
DO 290 J=1,ISS
GO TO (270,280), ISCLN
SAVEC(J)=GJA(J)*CMA
GO TO 290
SAVEC(J)=GJ(J)*CM
CONTINUE
IF (SKIP) GO TO 360
DETERMINE X FROM EFFECTIVE AREA RATIO
IF (ISW3B.EQ.0) GO TO 310
IF (ISONIC) 310,320,300
IF (SWATCH) GO TO 340

```

310 C 310 IS .CH=ISWTC+1
 IF (ISWTC.EQ.3) GO TO 330
 X BASED ON APPROXIMATING GEOMETRIC AREA RATIO BY AFNX
 CALL FINDX (AFNX,FLAG,CX)
 IF (ERR) RETURN
 IF (ISW3B.EQ.0) GO TO 410
 IF (CX.LT.XZERO) GO TO 180
 GO TO 370
 320 CX=C.
 DELTI=DELTSV
 DELTV=DELT1
 GO TO 370
 330 SWTC=.TRUE.
 XPB=XO
 XPB=XO-XP
 X CORRECTED FOR DISPLACEMENT THICKNESS
 CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)
 IF (ERR) RETURN
 IF (MODLPT) GO TO 360
 IF (CX.GT.XLAST) GO TO 370
 WRITE (6,640)
 IF (ISW3B.EQ.0.OR.AR.GT.4.*AFNX) GO TO 350
 IRSTRT=IRSTRT+1
 IF (IRSTRT.GT.3) GO TO 350
 W=W/2.
 WRITE (6,650) W
 GO TO 130
 350 WRITE (6,FRDMP)
 RETURN
 360 CALL TRANSP (CT*CTAP,PRES*PRESA)
 GO TO 410
 370 XLAST=CX
 CALL BLAYER (.TRUE.)
 IF (ERR) RETURN
 IF (IPASS.EQ.1.AND.ISONIC.NE.0) GO TO 180
 IF (ISW3B.EQ.0) GO TO 400
 DO 390 M=1,NPRFLS
 QMDS1=. .-DELBL(M)
 IF (IPASS.EQ.1) RETURN
 CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)
 IF (MODLPT) GO TO 500
 IF (.NOT.SKIP) GO TO 420
 SKIP=.FALSE.
 GO TO 550
 420 XP=XO
 XO=CX
 IF (ISW3B.EQ.0) GO TO 440
 DO 430 M=1,NPRFLS
 DELBLP(M)=DELBL(M)
 DELRLO(M)=DELBL(M)
 IF (CX.LT.XMAX) GO TO 450
 XMODEL=CX*XX
 LASTPT=.TRUE.
 IF (CX.GE.XMODEL) GO TO 460
 IF (SKIP) GO TO 220

ORIGINAL PAGE IS
 OF POOR QUALITY

```

460      GO TO 550
      X1=XP
      X2=XO
      MODLPT=.TRUE.
      SKIP=.FALSE.
      NMPI=0
      CT1=CT+DELT
      CT2=CT
      CTSAVE=CT
      IF (ISW3B.EQ.0) GO TO 490
      CXSAVE=CX
      XFRAC=(XMODEL-XP)/(XO-XP)
      DO 470 M=1,NPRFLS
      DELBL(M)=(DELBLC(M)-DELBLP(M))*XFRAC+DELBLP(M)
470      DO 480 K=1,13
      BLVARS(K)=BLVAR(K)
480      CT=CT1+(CT2-CT1)*(XMODEL-X1)/(X2-X1)
490      GO TO 220
      NMPI=NMPI+1
      ERRX=CX-XMODEL
      IF (ABS(ERRX).LE.0.0025) GO TO 520
      IF (NMPI.LE.9) GO TO 510
      WRITE (6,660) CX,XMODEL
      GO TO 520
      CT1=CT2
      CT2=CT
      X1=X2
      X2=CX
510      GO TO 490
      IF (LASTPT) GO TO 530
      CALL NEXTMP (ITS,XMOD1,XMODEL)
520      SKIP=.TRUE.
      IF (ISW3B.EQ.0) GO TO 550
      DO 540 K=1,13
      BLVAR(K)=BLVARO(K)+(BLVARS(K)-BLVARO(K))*XFRAC
540      CALL OUT2 (ISOLN)
550      IF (ERR) RETURN
      IF (.NOT.MODLPT) GO TO 580
      MODLPT=.FALSE.
      CALL MDEL
      IF (LASTPT) GO TO 610
      IF (ERR) RETURN
      CT=CTSAVE
      CX=CXSAVE
      IF (ISW3B.EQ.0) GO TO 440
      DO 560 K=1,13
      BLVAR(K)=BLVARS(K)
560      DO 570 M=1,NPRFLS
      DELBL(M)=DELBLO(M)
570      GO TO 440
      IF (ISW3B.EQ.0) GO TO 600
      DO 590 K=1,13
      BLVARO(K)=BLVAR(K)
580      IF (CX.GT.CXMAX) GO TO 610
590      600

```

IF (ISONIC,NE,0) GO TO 180	FRD 331
CT=CTSTAR	FRD 332
ISONIC=1	FRD 333
FLAG=1,	FRD 334
GO TO 180	FRD 335
IF (NPRFLS,EO,1) GO TO 620	FRD 336
AXISYM=.TRUE.	FRD 337
CALL MDEL	FRD 338
AXISYM=.FALSE.	FRD 339
RETURN	FRD 340
	FRD 341
	FRD 342
	FRD 343
FORMAT (1H1,3A4,8HSOLUTION//)	FRD 344
FORMAT (2PHOX DECREASED IN FROZED)	FRD 345
FORMAT (103H RESTART SOLUTION WITH A LARGER AVERAGING DISTANCE FOR	FRD 346
1 THE BOUNDARY LAYER CORRELATION WITH PARAMETER. NEW W=F9.6/1H1)	FRD 347
FORMAT (73H ITERATION TO OBTAIN FREE-STREAM SOLUTION AT MODEL POIN	FRD 348
IT DID NOT CONVERGE,5X,2HX=.1PE10.3,5X,7HXMODEL=.E10.3,3H CM)	FRD 349
END	FRD 350-

ORIGINAL PAGE IS
OF POOR QUALITY


```

56  GEO 56
57  GEO 57
58  GEO 58
59  GEO 59
60  GEO 60
61  GEO 61
62  GEO 62
63  GEO 63
64  GEO 64
65  GEO 65
66  GEO 66
67  GEO 67
68  GEO 68
69  GEO 69
70  GEO 70
71  GEO 71
72  GEO 72
73  GEO 73
74  GEO 74
75  GEO 75
76  GEO 76
77  GEO 77
78  GEO 78
79  GEO 79
80  GEO 80
81  GEO 81
82  GEO 82
83  GEO 83
84  GEO 84
85  GEO 85
86  GEO 86
87  GEO 87
88  GEO 88
89  GEO 89
90  GEO 90
91  GEO 91
92  GEO 92
93  GEO 93
94  GEO 94
95  GEO 95
96  GEO 96
97  GEO 97
98  GEO 98
99  GEO 99
100 GEO 100
101 GEO 101
102 GEO 102
103 GEO 103
104 GEO 104
105 GEO 105
106 GEO 106
107 GEO 107
108 GEO 108
109 GEO 109
110 GEO 110-

S1=1.
S2=C.
RHO=RHTH
GO TO 130
A2=S1*S1
N=3
OPHA=1.+0.5*A
IF (S2) 70,70,80
RHO=AMINI(RHTH+DELRHO,(1.-C/A2)**(1./A))
GO TO 90
RHO=AMAX1(RHTH-DELRHO, SORT(C)/S1)
N=N+1
IF (N-50) 100,100,130
CONTINUE
IF (RHO.GE.0.) GO TO 110
WRITE (6,200)
CALL DUMP (RNAME)
RETURN
Z=R1 **A
OMZ=1.-Z
AN=SQRT(C/OMZ)/RHO
DADR=AN/RHO*(OPHA*Z-1.)/OMZ
DD=(S1-AN)/DADR
RHO=RHO+DD
IF (ABS(DD/RHO).GE.1.0E-6) GO TO 90
Y=SIGN(1.0DD,CX*(RHO-RHTH))
IF (Y) 130,130,120
RHO=RHTH-SIGN(1.0D-02,CX)
N=0
GO TO 90
AFNX=SM/(SU*RHO)
IF (S2.NE.0.) GO TO 170
IF (NDRFLS.EQ.1) GO TO 150
D2LA=0.
DO 140 J=1,2
DELSTP(J)=RO*(1.-OMDST(J))
NSUJ=NSECTU(J)
D2LA=D2LA+1./ (RTHCM(J)-DELSTP(J))/PARAM(3,NSUJ,J)
GO TO 160
NSUJ=NSECTU(1)
D2LA=1./ (RTHCM(1)*OMDST(1)*PARAM(3,NSUJ,1))
IF (JDIM.NE.0) D2LA=2.*D2LA/SQRTA
DLOGR=-SQRT(C*D2LA/A/RHTH)
GO TO 190
DLOGR=((2.*C)/(A*(S1*RHO)**2-C*(A+2.)))*S2/S1
GO TO 190
AFNX=S1
DLOGA=S2/S1
RHO=SM/(SU*AFNX)
RETURN
C
C
C
200 FORMAT (21HNEGATIVE RHO IN GEOM)
END

```

```

SUBROUTINE GEOMAR (X,ARATIO,DERIVA)
  COMPUTES GEOMETRIC AREA RATIO AND ITS DERIVATIVE DERIVA.
  REAL ACOM(30),CLMENT(10),HP(20)
  DOUBLE PRECISION AA,CAPX,CDIJ
  COMMON AA(22,24),CDIJ(20,10),CAPX(20)
  COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO,
1 C, CARB, CH, CTA, CTAP, CTB, CMA, CRA, CRP,
2 CRRB, CRS, CSTA, CTI, CX, CXB, CMAX, DATEST, DBTEST, DELT1,
3 CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT1,
4 DELT2, DELTAX, DLOGA, DLOGR, PRESTH, DT, ENT, FLUX, HDELX, PCT,
5 SPECTEST, PRESA, RHOC, RHOP, RHPL, RHTH, ROBARR, RHO, RHOB,
6 GRHOBAR, RHOC, RHOP, RHPL, RHTH, ROBARR, RHO, RHOB,
7 SEN, SHPG, SC, SL, SL54, SM, SU, SU2, SUMG,
8 TEST, TESTB, TPRINT, TSTOP, UP, ZPA
  COMMON BE(64), RET(20), RLK(31), CAI(64), CAPXTH(20),
1 CCPJ(20), CFACT(64), CGI(20), CGMU(20), CHI(64),
2 CLNMC(64), CLNFI(64), CMW(20), ETAL(64), ETAJ(20), GJA(20),
3 GJB(20), PERTGJ(20), PGJ(20), PI(64), PCHI(64), QM(20),
4 QQ(64), SAJ(20), SDCHI(64), SFNT(20), SHJ(20), SHJA(20),
5 SKIL(20), SS(20), TB(30), TFA(20), TFC(20), TFI(20),
6 TFO(20), TFE(20), TFK(20), THEV(20), XMJAT(20), XNUI(64)
  COMMON BETA(64,20), ELJ(10,20), GELJ(10,20),
1 XNUIJ(64,20),XNUIJP(64,20)
  COMMON IC, IM, INEQ, INEQV, IP, IRUN, ISC, ISCP1,
1 ISMC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
3 ISW5B, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, M1, NFIT,
4 NIT, NNN, NNS, NOS, NQT, NTEST
  COMMON IGM(20), ITB(5), KUR(64,20), LPIJ(20,10)
  COMMON ACOM,ELMENT,HP
  COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),
1 NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL
  COMMON /BL/ DELFL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
1 SW,RC,JDIN,IPCINT
  DIMENSION SN(2),RAD(2),S1(2),S2(2),DRV(2)
  DATA NAME /6HGEOMAR/
  DATA SN /-1.,1./
  COMMON /THRT/ PSA
  ENTRY=1
  GO TO 10
  ENTRY GMAR(X,Y)
  ENTRY=2
  GO TO 10
  ENTRY GMAR2(X,Y,Z)
  ENTRY=3
  GO TO 10
  ENTRY GMAR3(X,OYDX,OZDX,Y,Z)
  ENTRY=4
  DO 100 I=1,NPRFLS
  NSM1=NSECT(I)-1
  DO 20 J=1,NSM1
  K=J
  IF (X,LT,ATPI(J,1)) GO TO 30
  CONTINUE
  K=NSECT(I)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

7  ISJ=ISHAPE(K,I)
   IF (ISJ.GT.1) GO TO 7C
   RAD(1)=PARAM(1,K,I)+PARAM(2,K,I)*X
   GO TO (50,100,100,40), IENTRY
40  DRV(1)=PARAM(2,K,I)
   GO TO 100
50  RATE=RAD(1)/RTHCM(I)
   IF (JDIM.EQ.0.OR.NPRFLS.EQ.2) GO TO 60
   S1(1)=RATE**2
   S2(1)=2.*RATE/RTHCM(I)*PARAM(2,K,I)
   GO TO 140
60  S1(1)=RATE
   S2(1)=PARAM(2,K,I)/RTHCM(I)
   GO TO 100
70  DAB=X-PARAM(2,K,I)
   Q=SQRT(PARAM(3,K,I)**2-DAB**2)
   RAD(1)=PARAM(1,K,I)+SN(1,ISJ-1)*Q
   GO TO (80,100,100,80), IENTRY
80  DRV(1)=-S1(1,ISJ-1)*DAB/Q
   IF (IENTRY.EQ.4) GO TO 100
   RATE=RAD(1)/RTHCM(I)
   IF (JDIM.EQ.0.OR.NPRFLS.EQ.2) GO TO 90
   S1(1)=RATE**2
   S2(1)=2.*RATE/RTHCM(I)*DRV(1)
   GO TO 100
90  S1(1)=RATE
   S2(1)=DRV(1)/RTHCM(I)
   CONTINUE
100 IF (NPRFLS.EQ.1) GO TO 130
   GO TO (120,130,110,110), IENTRY
110 SOTRSA=SQRT(RSA)
   Y=RAD(1)*SOTRSA
   Z=RAD(2)*SOTRSA
   IF (IENTRY.EQ.3) GO TO 180
   DYDX=DRV(1)*SOTRSA
   DZDX=DRV(2)*SOTRSA
   GO TO 180
120 ARATIO=S1(1)*S1(2)*RSA
   DERIVA=(S1(1)*S2(2)+S1(2)*S2(1))*RSA
   GO TO 100
130 IF (IENTRY.EQ.2) GO TO 160
140 ARATIO=S1(1)*RSA
   DERIVA=S2(1)*RSA
150 IF (ARATIO.LT.1..AND.ARATIO.GE.0.99999) ARATIO=1.
   IF (ARATIO.GE.1..AND.X*DERIVA.GE.0.) GO TO 180
   WRITE (6,190) X,ARATIO,DERIVA
   CALL DUMP (NAME)
   RETURN
160 Y=RAD(1)
   IF (JDIM.EQ.0) GO TO 170
   Y=Y*SQRT(RSA)
   GO TO 180
170 Y=Y*RSA
180 RETURN
C

```

```

GMR 56
GMR 57
GMR 58
GMR 59
GMR 60
GMR 61
GMR 62
GMR 63
GMR 64
GMR 65
GMR 66
GMR 67
GMR 68
GMR 69
GMR 70
GMR 71
GMR 72
GMR 73
GMR 74
GMR 75
GMR 76
GMR 77
GMR 78
GMR 79
GMR 80
GMR 81
GMR 82
GMR 83
GMR 84
GMR 85
GMR 86
GMR 87
GMR 88
GMR 89
GMR 90
GMR 91
GMR 92
GMR 93
GMR 94
GMR 95
GMR 96
GMR 97
GMR 98
GMR 99
GMR 100
GMR 101
GMR 102
GMR 103
GMR 104
GMR 105
GMR 106
GMR 107
GMR 108
GMR 109
GMR 110

```

GMR 111
GMR 112
GMR 113
GMR 114
GMR 115-

FORMAT (40H)ERROR IN INPUT DATA FOR NOZZLE GEOMETRY.4H
1.9H ARATIO=.E14.6.9H DERIVA=.E14.6)
END

ORIGINAL PAGE IS
OF POOR QUALITY


```

40      BET(I)=-1.0
50      DO 40 K=1,ISC
60      BET(I)=BET(I)+CDIJ(I,K)
      CONTINUE
      SUM=0.0
      DO 80 I=1,ISC
      OM(I)=0.0
      DO 70 K=1,ISC
      OM(I)=OM(I)+AIN(K,I)*CAPQ(K)
      SUM=SUM+OM(I)
      CONTINUE
      DO 90 I=1,ISC
      OM(I)=OM(I)/SUM
      CCI(I)=0.0
      DO 90 J=1,ISC
      CCI(I)=CCI(I)+LPIJ(I,J)*CMW(J)
      DO 100 I=1,ISC
      CGI(I)=CCI(I)
      IF (ISMC.EQ.0) GO TO 130
      DO 120 I=1,ISMC
      SUM=0.0
      DO 110 K=1,ISC
      SUM=SUM+CDIJ(I,K)*CGI(K)
      IL=I+ISC
      CGI(IL)=SUM
      CONTINUE
      RETURN
      ENTRY INTA
      CALL EQCALC (CTAP,PRESA)
      IF (ERR) RETURN
      ZPA=ZPZ
      CM=ZCM
      SEN=ZSEN
      CHA=ZCH
      CMA=CM
      CRR=SEN/CMA
      SEN=CRR*CR
      CHE=CHA
      DO 140 I=1,ISS
      CGNU(I)=ZGMU(I)
      CAPX(I)=ZCAP(I)
      GJA(I)=CAPX(I)/CMA
      ROBARP=ZRP
      RHAP=ZRHQ
      ROBARA=ROBARP/RHAP
      RETURN
      END

```

. 140

```

SUBROUTINE INIT
REAL ACOX(30),ELMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A, AFNTS, AFNX, AMACH,
1 C, CARB, CH,
2 CRR, CRS, CSTA, CT, CTAP,
3 CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT,
4 DELT, DELTAX, CLOGA, OLOGA, PRESB, PRESTH, PRHC, RHAP, RHOB,
5 PCTEST, PRES, PRESA, RHOC, RHOP, RHPL, RHTR, ROBAR, ROBRAP, SCPP,
6 PHOBAR, RHOC, RHOP, RHPL, RHTR, ROBAR, ROBRAP, SCPP,
7 SEN, SHPG, SC, SL, SL64, SM, SU, SUMG,
8 TEST, TESTB, TPRINT, TSTOP, UP, ZPA
COMMON RE(64), BFB(20), BLBK(31), CAI(64), CAPXTH(20),
1 CCRPJ(20), CEACT(64), CGI(20), CGU(20), CHI(64), CHI(20),
2 CLN1VC(64), CLNPI(64), CMW(20), ETAI(64), FTAJ(20), GJA(20),
3 GUT(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), QM(20),
4 GJ(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20),
5 SKIL(20), SS(20), TH(30), TFA(20), TFB(20), TFC(20),
6 TFE(20), TFK(20), THEV(20), XMJAT(20), XNUI(64)
COMMON RETA(64,20), ELJ(10,20), GELJ(10,20),
1 XNUIJ(64,20), XNUIJP(64,20)
COMMON IC, IM, INEQ, IP, IRUN, ISC, ISCP1,
1 ISMC, ISVCR, ISR, ISS, ISSNP, ISSP2, ISSP3, ISSP4,
2 ISK1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
3 ISW5B, ISW6A, ISW6B, IUPC, IZER, JJK, LC, MI, NFIT,
4 NIT, NNN, NNS, NOS, NQT, NTEST
COMMON IGJ(20), IGM(20), ITR(5), KUR(64,20), LPIJ(20,10)
COMMON ACOV,ELMENT,HP
COMMON ACOV,ELMENT,HP
DIMENSION AAA(22,24),RTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SB
1 J(20),SDGJ(20),SHJAP(20),THEVP(20)
EQUIVALENCE (AAA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),
1 (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SHJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
COMMON /MASSFL/ SMAS,CTMXXI,TSTOP1,IS(20)
DIMENSION SPRB(43,30)
COMMON /POLYAT/ THLVE(4,20)
COMMON /SPEC/ SP1(43),SP2(43),SP3(43),SP4(43),SP5(43),SP6(43),
1 SP7(43),SP8(43),SP9(43),SP10(43),SP11(43),SP12(43),SP13(43),
2 SP14(43),SP15(43),SP16(43),SP17(43),SP18(43),SP19(43),
3 SP20(43),SP21(43),SP22(43),SP23(43),SP24(43),SP25(43),
4 SP26(43),SP27(43),SP28(43),SP29(43),SP30(43)
EQUIVALENCE (SP1(1),SPRP(1,1))
COMMON /TNONEQ/ CTE,DTE,RPAR,EPAR(2,25),NT,ITR(25),KTF(25),
1 KTR(25),ICH(20),IPA(25)
CTMXX=CTMXXI/CTAP
IF (INEOV.NE.0) CTMXX=2.
TSTOP=TSTOP1/CTAP
IF (TSYCP.EQ.0.) TSTOP=0.004
ARRA=0.0
ARRB=0.0
DATEST=1.001
OBTST=5.0
PCTEST=1.2

```


**ORIGINAL PAGE IS
OF POOR QUALITY**

-456-

1	KAN	SUBROUTINE KANDMU (ZK)
2	KAN	SUBROUTINE KANDMU COMPUTES VISCOSITY AND TRANSLATIONAL COMPONENT
3	KAN	OF THERMAL CONDUCTIVITY. INDEX L=1 INDICATES TRANSLATIONAL
4	KAN	THERMAL CONDUCTIVITY, L=2 INDICATES VISCOSITY.
5	KAN	DIMENSION ZK(2)
6	KAN	COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
7	KAN	COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)
8	KAN	COMMON /TRANS5/ N,IELEC,IDI,ID2
9	KAN	COMPUTE DIAGONAL MATRIX ELEMENTS CAP-A(I)
10	KAN	DO 20 I=1,N
11	KAN	A(I,1)=C.
12	KAN	DO 10 J=1,N
13	KAN	B(I,1)=B(I,1)+X(J)*Q(3,I,J)
14	KAN	B(I,2)=B(I,2)/ZM2(I)
15	KAN	BR(I,1)=1./B(I,1)
16	KAN	BR(I,2)=1./B(I,2)
17	KAN	IF (N.EQ.1) GO TO 70
18	KAN	COMPUTE AVERAGE OF NON-DIAGONAL MATRIX ELEMENTS SMALL-A(I,J)
19	KAN	DO 60 L=1,2
20	KAN	U=0.
21	KAN	Y=0.
22	KAN	DO 40 I=2,N
23	KAN	II=I-1
24	KAN	V=0.
25	KAN	W=0.
26	KAN	DO 30 J=1,II
27	KAN	R=X(J)*(BR(I,L)-BR(J,L))*2
28	KAN	V=V+R
29	KAN	W=W+Q(L,I,J)
30	KAN	U=U+X(II)*W
31	KAN	Y=Y+X(II)*V
32	KAN	IF (Y.EQ.0.) GO TO 50
33	KAN	A(II)=U/Y
34	KAN	GO TO 50
35	KAN	A(L)=C.
36	KAN	CONTINUE
37	KAN	A(II)=5*A(II)
38	KAN	COMPUTE VISCOSITY AND TRANSLATIONAL THERMAL CONDUCTIVITY
39	KAN	DO 90 L=1,2
40	KAN	Y=0.
41	KAN	DO 80 I=1,N
42	KAN	Y=Y+X(I)/(A(L)+B(I,L))
43	KAN	ZK(L)=3.125*Y/(1.-A(L))*Y
44	KAN	RETURN
45	KAN	END

```

SUBROUTINE KINT (ZKINT)
COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
COMMON /TRANS3/ B(20,2),BR(20,2),A(2),X(20),DH(20)
COMMON /TRANS5/ N,I,ELEC,I01,I02
COMPUTE SUM OVER J OF X(J)*DELTA(I,J)
IF (N.EQ.1) GO TO 20
N1=N+1
DO 10 I=2,N
  I1=I-1
  N1=N-I1
  DO 10 J=1,I1
    NJ=N1-J
    DO 10 L=1,2
      B(I,L)=B(I,L)+X(J)*Q(L,J,I)
      B(N1,L)=B(N1,L)+X(NJ)*Q(L,N1,NJ)
COMPUTE THERMAL CONDUCTIVITY COMPONENT DUE TO INTERNAL EXCITATION
ENERGY
ZKINT=0.
DO 30 I=1,N
  ZKINT=ZKINT+DH(I)*X(I)/B(I,1)
RETURN
END

```

1 KIN
 2 KIN
 3 KIN
 4 KIN
 5 KIN
 6 KIN
 7 KIN
 8 KIN
 9 KIN
 10 KIN
 11 KIN
 12 KIN
 13 KIN
 14 KIN
 15 KIN
 16 KIN
 17 KIN
 18 KIN
 19 KIN
 20 KIN
 21 KIN
 22 KIN

ORIGINAL PAGE IS
 OF POOR QUALITY

[illegible]

[illegible]

```

150 140 I=1,ISR
160 WRITE (6,570) I,KTF(I),KTR(I),ITR(I),EPAR(1,I),EPAR(2,I),IPA(I)
170 IF (NFI) 160,190,160
180 WRITE (6,510)
190 DO 180 J=1,ISS
200 IF (IGJ(J)) 170,180,170
210 WRITE (6,520) HP(J),TFA(J),TFC(J),TFD(J),TFE(J),TFK(J),SHJA
220 1P(J)
230 CONTINUE
240 IF (INEQV.NE.0) WRITE (6,580)
250 IF (INFT) 160,190,160
260 WRITE (6,590) HP(JATOM),HP(JMOL),(COM1(I,IAMBIP),I=1,2),(COM2(K,LE
270 1WIS),K=1,2)
280 IF (ISW3H.NE.0) GO TO 200
290 WRITE (6,530)
300 GO TO 210
310 WRITE (6,540)
320 IF (XMODPI.GF.9.E19.AND.NTS.EQ.0) GO TO 340
330 WRITE (6,600)
340 CXMAXI=CXMAX/2.54
350 IF (XMODPI.LT.9.E19) WRITE (6,630) NMODPT,XMODPI,CXMAXI
360 IF (NTS.EQ.0) GO TO 240
370 GO TO (220,230),NPFPLS
380 WRITE (6,610) (TSDIAM(I),I=1,NTS)
390 GO TO 240
400 WRITE (6,62) (TSDIAM(I),I=1,NTS)
410 IF (T.AXISYM) GO TO 290
420 WRITE (6,640) TMODEL,TPLATE
430 IF (FSIAG) 250,260,270
440 WRITE (6,650)
450 GO TO 280
460 WRITE (6,660)
470 GO TO 280
480 WRITE (6,670)
490 WRITE (6,680) CATFAC
500 KDPI=KDIM+1
510 WRITE (6,690) (WORD(I,KDPI),I=1,3)
520 IF (.NOT.WEDGEN) GO TO 340
530 WRITE (6,700) (ANGLE(I),I=1,NANGLE)
540 WRITE (6,710) (RADLE(I),I=1,NRAOLE)
550 IF (NWX.EQ.0) GO TO 300
560 WRITE (6,720) NWX,DWX,NWXI
570 IF (WXI(I).GT.1.E19) GO TO 330
580 DO 310 I=1,20
590 IF (WXI(I).GT.1.E19) GO TO 320
600 NWXI=I
610 CONTINUE
620 WRITE (6,730) (WXI(I),I=1,NWXI)
630 WRITE (6,740) TWEDGE,WK
640 RETURN
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

380	FORMAT (1H, //122H REACTION NO.	INSTANT FACTOR A1	TEMP.	LIS 166
390	1POWER DEPENDENCE	CHI TEST	THIRD BOO	LIS 167
400	2Y MATR (1X, //122H REACTION NO.			LIS 168
410	FORMAT (1H, //122H REACTION NO.			LIS 169
420	FORMAT (1H, //122H REACTION NO.			LIS 170
430	FORMAT (1H, //122H REACTION NO.			LIS 171
440	FORMAT (1H, //122H REACTION NO.			LIS 172
450	FORMAT (1H, //122H REACTION NO.			LIS 173
460	FORMAT (1H, //122H REACTION NO.			LIS 174
470	FORMAT (1H, //122H REACTION NO.			LIS 175
	FORMAT (1H, //122H REACTION NO.			LIS 176
	FORMAT (1H, //122H REACTION NO.			LIS 177
	FORMAT (1H, //122H REACTION NO.			LIS 178
	FORMAT (1H, //122H REACTION NO.			LIS 179
	FORMAT (1H, //122H REACTION NO.			LIS 180
480	FORMAT (1H, //122H REACTION NO.			LIS 181
490	FORMAT (1H, //122H REACTION NO.			LIS 182
500	FORMAT (1H, //122H REACTION NO.			LIS 183
510	FORMAT (1H, //122H REACTION NO.			LIS 184
	FORMAT (1H, //122H REACTION NO.			LIS 185
	FORMAT (1H, //122H REACTION NO.			LIS 186
520	FORMAT (1H, //122H REACTION NO.			LIS 187
530	FORMAT (1H, //122H REACTION NO.			LIS 188
540	FORMAT (1H, //122H REACTION NO.			LIS 189
550	FORMAT (1H, //122H REACTION NO.			LIS 190
560	FORMAT (1H, //122H REACTION NO.			LIS 191
	FORMAT (1H, //122H REACTION NO.			LIS 192
	FORMAT (1H, //122H REACTION NO.			LIS 193
	FORMAT (1H, //122H REACTION NO.			LIS 194
570	FORMAT (1H, //122H REACTION NO.			LIS 195
580	FORMAT (1H, //122H REACTION NO.			LIS 196
590	FORMAT (1H, //122H REACTION NO.			LIS 197
600	FORMAT (1H, //122H REACTION NO.			LIS 198
610	FORMAT (1H, //122H REACTION NO.			LIS 199
	FORMAT (1H, //122H REACTION NO.			LIS 200
	FORMAT (1H, //122H REACTION NO.			LIS 201
	FORMAT (1H, //122H REACTION NO.			LIS 202
620	FORMAT (1H, //122H REACTION NO.			LIS 203
630	FORMAT (1H, //122H REACTION NO.			LIS 204
640	FORMAT (1H, //122H REACTION NO.			LIS 205
650	FORMAT (1H, //122H REACTION NO.			LIS 206
660	FORMAT (1H, //122H REACTION NO.			LIS 207
670	FORMAT (1H, //122H REACTION NO.			LIS 208
	FORMAT (1H, //122H REACTION NO.			LIS 209
	FORMAT (1H, //122H REACTION NO.			LIS 210
	FORMAT (1H, //122H REACTION NO.			LIS 211
680	FORMAT (1H, //122H REACTION NO.			LIS 212
690	FORMAT (1H, //122H REACTION NO.			LIS 213
700	FORMAT (1H, //122H REACTION NO.			LIS 214
710	FORMAT (1H, //122H REACTION NO.			LIS 215
720	FORMAT (1H, //122H REACTION NO.			LIS 216
730	FORMAT (1H, //122H REACTION NO.			LIS 217
	FORMAT (1H, //122H REACTION NO.			LIS 218
740	FORMAT (1H, //122H REACTION NO.			LIS 219

IRAG COEFFICIENT=F5.3)
END

LIS 221
LIS 222-

ORIGINAL PAGE IS
OF POOR QUALITY


```

C      SUBROUTINE MATINV (A,MM,NN)
C      MATRIX INVERSION
C      INVERSE REPLACES ORIGINAL MATRIX
C      CALLING SEQUENCE
C      MM - SIZE TO BE INVERTED
C      NN - FIRST SUBSCRIPT OF A IN DIMENSION
C
C      DOUBLE PRECISION A
C      DATA RNAME /6HMATINV/
C      DIMENSION A(NN,10),IV(10)
C      M=MM
C      DO 10 I=1,M
C      IV(I)=I
C      JA=1
C      JJ=JA
C      K=JJ
C      KI=K+1
C      BIG=ABS(A(K,K))
C      IF (KI-M) 30,30,60
C      DO 50 I=KI,M
C      IF (BIG-ABS(A(I,JJ))) 40,50,50
C      K=I
C      BIG=ABS(A(I,JJ))
C      CONTINUE
C      BIG=A(K,JJ)
C      IF (BIG) 80,70,20
C      CONTINUE
C      WRITE (6,300)
C      CALL DUMP (RNAME)
C      RETURN
C      IF (K-JJ) 90,110,90
C      J=IV(K)
C      IV(K)=IV(JJ)
C      IV(JJ)=J
C      DO 100 J=1,M
C      B=A(JJ,J)
C      A(JJ,J)=A(K,J)
C      A(K,J)=B
C      A(JJ,JJ)=1.0
C      JA=JJ+1
C      DO 120 J=1,M
C      A(JJ,J)=A(JJ,J)/BIG
C      JB=JJ-1
C      IF (JB) 170,170,130
C      DO 160 I=1,JB
C      BB=A(I,JJ)
C      A(I,JJ)=0.0
C      IF (BB) 140,160,140
C      DO 150 J=1,M
C      A(I,J)=A(I,J)-BB*A(JJ,J)
C      CONTINUE
C      IF (JA-M) 180,180,220
C      DO 210 I=JA,M
C      BB=A(I,JJ)
C      A(I,JJ)=0.0

```

```

10 MAT
11 MAT
12 MAT
13 MAT
14 MAT
15 MAT
16 MAT
17 MAT
18 MAT
19 MAT
20 MAT
21 MAT
22 MAT
23 MAT
24 MAT
25 MAT
26 MAT
27 MAT
28 MAT
29 MAT
30 MAT
31 MAT
32 MAT
33 MAT
34 MAT
35 MAT
36 MAT
37 MAT
38 MAT
39 MAT
40 MAT
41 MAT
42 MAT
43 MAT
44 MAT
45 MAT
46 MAT
47 MAT
48 MAT
49 MAT
50 MAT
51 MAT
52 MAT
53 MAT
54 MAT
55 MAT

```

```

190 IF (BB) 190,210,190
200 DO 200 J=1,M
210 A(I,J)=A(I,J)-BB*A(JJ,J)
220 CONTINUE
230 GO TO 20
240 M=M-1
250 IF (M) 290,290,230
260 DO 280 J=1,M1
270 DO 250 I=J,M
280 IF (IV(I)-J) 250,240,250
290 K=I
300 GO TO 260
310 CONTINUE
    WRITE (6,310)
    CALL DUMP (RNAME)
    RETURN
260 IV(K)=IV(J)
    DO 270 L=1,M
    AA=A(L,J)
    A(L,J)=A(L,K)
    A(L,K)=AA
    CONTINUE
    RETURN
    FORMAT (1H1,26H MATINV, MATRIX SINGULAR /)
    FORMAT (1H1,37HINDEXING OR STORAGE FAILURE IN MATINV/)
    END

```

```

MAT 56
MAT 57
MAT 58
MAT 59
MAT 60
MAT 61
MAT 62
MAT 63
MAT 64
MAT 65
MAT 66
MAT 67
MAT 68
MAT 69
MAT 70
MAT 71
MAT 72
MAT 73
MAT 74
MAT 75
MAT 76
MAT 77
MAT 78
MAT 79
MAT 80
MAT 81
MAT 82
MAT 83-

```

```

SUBROUTINE MODEL
CALCULATES STAGNATION CONDITIONS BEHIND STRONG NORMAL SHOCK AND
CONDITIONS ON MODEL
LOGICAL LO,HI,DATAPE,WEDGEM,AXISYM,ERR,SUPGO,NOTRAN
REAL ET,ETO
DOUBLE PRECISION AA,CAPX,CDIJ,GJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFNX,AMACH,AR,ARBA,ARBB,BZERO,
C,CARB,CRS,CSTA,CT,CTAP,CTB,CMA,CTC,CTMAX,CTMXX,
1 CTP,CTPL,CTT,CX,CXB,CXMAX,DATEST,DBTEST,DELT1,
2 CRRB,CRS,CSTA,CT,CTAP,CTB,CMA,CTC,CTMAX,CTMXX,
3 CTP,CTPL,CTT,CX,CXB,CXMAX,DATEST,DBTEST,DELT1,
4 DELT2,DELTAX,DLOGA,DLOGR,DLOGR,PRESA,PRESB,PRESH,PRHO,RHAP,
5 SPCTEST,PHOC,RHOP,RHPL,RHTR,ROBARA,ROBARP,SCPG,SDT,
6 RHOBAR,PHOC,RHOP,RHPL,RHTR,ROBARA,ROBARP,SCPG,SDT,
7 SEN,SHPG,SC,SL,SL64,SM,SU,SUMG,
8 TEST,TESTB,TPRINT,TSTOP,UP,ZP,ZPA
9 COMMON BE(64),BET(20),BLBK(31),CAI(64),CAPXTH(20),
10 CCPJ(20),CEACT(64),CGI(20),CGMU(20),CH(64),CHII(20),
11 CLNIMC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),GJA(20),
12 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
13 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
14 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
15 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
16 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
17 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
18 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
19 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
20 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
21 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
22 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
23 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
24 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
25 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
26 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
27 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
28 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
29 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
30 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
31 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
32 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
33 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
34 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
35 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
36 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
37 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
38 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
39 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
40 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
41 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
42 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
43 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
44 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
45 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
46 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
47 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
48 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
49 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
50 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
51 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
52 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
53 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
54 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
55 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20)

```

```

COMMON /RDMOD/ LEWIS,IAMBIP
COMMON /COLDSP/ CGMW,QPJ(10),IJCS(10),NCS
COMMON /RDMAIN/ HS,SUPGC,MFILTER,NOTRAN
DIMENSION QUANT(20,2)
DATA QUANT /3HT2E,4HRT2E,4HPT2E,5HMT2E,5HWT2E,
6HSGT2E,3HPRF,5HPSLE,4HQEFF,5HDELRFF,
6HDELREF,4HQEFF,4HQEFF,5HHRATE,3HQFP,3HT2F,4HRT2F,
4HPT2F,5HMT2F,3HLEF,4HST2F,5HWT2F,6HSGT2F,3HPRF,5HPSLF,
4HQEFF,4HQEFF,5HPSLE,6HDELREF,4HQEFF,4HQEFF,
5HKT2FF,5HHRATE,1H /
DIMENSION TITLE2(3,2)
DATA TITLE2 /6HEQUILI,6HBRIUM ,5HSHOCK,6H FROZEN,6H SHOCK,1H /
DATA EQUALS /2H =/
DATA CATM /1.01325F6/
DATA SQP /C.747.1./
NAMELIST /MODDMP/T2E,EPSON,T2F,EPSP,PS,ZCH,ZCM,ZRHO,HS,R1,P1,
1 ICOUNT,LO,HI,ERZ,ERRLO,ERRHI,TLO,T2,P2,EPSON,EPSOND,H2,
2 ALPHA,T5,DISC
DIMENSION SCOUT(20)
IF (.NOT.(AXISYM.OR.WEDGEM)) GO TO 480
IF (AMACH.GE.1.5) GO TO 10
WRITE (6,490)
GO TO 480
CONTINUE
NORMAL SHOCK SOLUTION
CALL ELTIME (ETO,0)
UI=SORT(8,31436E7*CTAP/CMA)*SU
R1=RHAP*RH0
HD=0.5*SU**2
HS=HD+CH
R1UI2=R1*UI**2/CATM
P1=PPRESA*PRES
PT=P1+R1UI2
PC=4,*R1*R1UI2
TI=CTAP*CT
CTSAVE=CT
CM1=82.055*R1*T1/P1
IF (EPSON.GT.0.) GO TO 40
EPSON=0.01
IF (NDTRAN) GO TO 40
DO 30 I=1,ISS
DO 20 K=1,NCS
IF (IJCS(K).NE.1) GO TO 20
SAVEC(I)=QPJ(K)
GO TO 30
CONTINUE
SAVEC(I)=0.
CONTINUE
CALL TRANSP (TMODEL,PRESA)
PRW=PR
WRITE (6,500) PRW
IF (EPSP.EQ.0.) EPSF=0.01
DO 440 ISOLN=1,2
IF (ISOLN.EQ.1.AND..NOT.AXISYM) GO TO 440
IF (FSTAG,LT.0.0.AND.ISOLN.EQ.1) GO TO 440

```

ORIGINAL PAGE IS
OF POOR QUALITY

10
C

20

30

40

MOD 56
MOD 57
MOD 58
MOD 59
MOD 60
MOD 61
MOD 62
MOD 63
MOD 64
MOD 65
MOD 66
MOD 67
MOD 68
MOD 69
MOD 70
MOD 71
MOD 72
MOD 73
MOD 74
MOD 75
MOD 76
MOD 77
MOD 78
MOD 79
MOD 80
MOD 81
MOD 82
MOD 83
MOD 84
MOD 85
MOD 86
MOD 87
MOD 88
MOD 89
MOD 90
MOD 91
MOD 92
MOD 93
MOD 94
MOD 95
MOD 96
MOD 97
MOD 98
MOD 99
MOD 100
MOD 101
MOD 102
MOD 103
MOD 104
MOD 105
MOD 106
MOD 107
MOD 108
MOD 109
MOD 110

```

50 EPS=SHV(2,ISOLN)
   T2=SHV(1,ISOLN)
   ICOUNT=0
   LO=.FALSE.
   HI=.FALSE.
   IF (ISOLN.EQ.1) GO TO 70
80 DO 60 I=1,ISS
   SAVEC(I)=GJ(I)*CM1
   GO TO 110
70 P2=P1 J12*(1.-EPS)
   ICOUNT=ICOUNT+1
   IF (ICOUNT.GT.30) GO TO 450
   CALL EQCALC (T2,P2)
   IF (F4R) RETURN
   EPSOLD=EPS
   EPS=R1/ZRHO
   IF (ABS(EPS/EPSOLD-1.).LE.0.001) GO TO 130
   ALPHA=P2/ZRHO
   DISC=PT*2-PC*ALPHA
   IF (DISC.GE.0.) GO TO 100
   EPS=EPSOLD
   IF (LU) GO TO 90
   T2=0.9*T2
   GO TO 50
90 T2=0.5*(T2+TLO)
   GO TO 50
100 P2=0.5*(PT+SQRT(DISC))
   GO TO 80
110 ICOUNT=ICOUNT+1
   CT=T2/CTAP
   IF (ICOUNT.GT.30) GO TO 450
   CALL THERM
   ZCM=CM1
   ZCH=0.
   DO 120 I=1,ISS
   ZCH=ZCH+SAVEC(I)*SHJ(I)
   OP=P1*T2/T1
   DISC=PT*2-4.*R1U12*OP
   IF (DISC.LT.0.) GO TO 450
   P2=0.5*(PT+SQRT(DISC))
   R2=R1*P2/OP
   EPS=R1/R2
130 H2=HS-HC*EPS**2
   ZCH=ZCH+ZCM/ZCM
   IF (ABS(ZCH/H2-1.).LE.0.001) GO TO 170
   ERZ=ZCH-H2
   IF (ERZ) I40,I4C,I50
   FRRLO=FRZ
   TLO=T2
   LO=.TRUE.
   IF (HI) GO TO 160
   T2=1.1*T2
   GO TO 50
150 FRRHI=ERZ
   THI=T2

```

```

MOD 111
MOD 112
MOD 113
MOD 114
MOD 115
MOD 116
MOD 117
MOD 118
MOD 119
MOD 120
MOD 121
MOD 122
MOD 123
MOD 124
MOD 125
MOD 126
MOD 127
MOD 128
MOD 129
MOD 130
MOD 131
MOD 132
MOD 133
MOD 134
MOD 135
MOD 136
MOD 137
MOD 138
MOD 139
MOD 140
MOD 141
MOD 142
MOD 143
MOD 144
MOD 145
MOD 146
MOD 147
MOD 148
MOD 149
MOD 150
MOD 151
MOD 152
MOD 153
MOD 154
MOD 155
MOD 156
MOD 157
MOD 158
MOD 159
MOD 160
MOD 161
MOD 162
MOD 163
MOD 164
MOD 165

```

```

160 HI=.TRUE.
170 IF (LO) GO TO 160
    T2=0.9*T2
    GO TO 50
    T2=T2-ERZ*(THI-TLO)/(ERRHI-ERRLO)
    GO TO 50
    C SUPERSONIC FLOW FROM THE SHOCK TO THE STAGNATION POINT ON THE MODEL
170 PS=PS+C*.5*RIU12*EPS
    TS=T2
    HI=.FALSE.
    ICOUNT=0
    ICOUNT=ICOUNT+1
    IF (ICOUNT.GT.20) GO TO 450
    GO TO (190,200).ISOLN
    CALL EQCALC (TS,PS)
    IF (ERR) RETURN
    GO TO 220
    CT=TS/CTAP
    CALL THERM
    ZCH=0.
    DO 210 I=1,ISS
    ZCH=ZCH+SAVEC(I)*SHJ(I)
    ZCH=ZCH+CMA/ZCM
    IF (ABS(ZCH/HS-1.)*LE.0.001) GO TO 260
    ERZ=ZCH-HS
    IF (ERZ) 230,230,240
    TLO=TS
    ERRLC=ERZ
    IF (HI) GO TO 250
    TS=1.1*TS
    GO TO 180
    THI=TS
    ERRLH=ERZ
    HI=.TRUE.
    TS=TS-ERZ*(THI-TLO)/(ERRHI-ERRLO)
    GO TO 180
    IF (ISOLN.EQ.1) GO TO 280
    IF (.NOT.AXISYM.OR.FSTAG.EQ.0.) GO TO 300
    ZSEN=-ALOG(PS)
    DO 270 I=1,ISS
    IF (SAVEC(I)*LE.0.) GO TO 270
    ZSEN=ZSEN+SAVEC(I)*(SENT(I)-ALOG(SAVEC(I)))
    CONTINUE
    ZRHO=2*PS/P2*T2/TS
    C VGP(1) = VELOCITY GRADIENT PARAMETER FOR HEMISPHERE MODEL
    C VGP(2) = VELOCITY GRADIENT PARAMETER FOR FLAT-FACED MODEL
    C SOD(1) = STANCOFF DISTANCE FOR HEMISPHERE (CYLINDER IF KDIM=0)
    C SOD(2) = STANCOFF DISTANCE FOR FLAT-FACED MODEL
    VGP(1)=SQRT(2.*PS*CATM*(1.-PI/PS)/ZRHO)/U1
    SOTE=SQRT(EPS)
    IF (KDIM.EQ.0.) GO TO 290
    SOD(1)=0.78*EPS
    IF (EPS.GT.C.1901) SOD(1)=SOD(1)*(1.+3.5*(EPS-0.19)**1.6)
    VCP(2)=EPS/(1.04+4.2*(SQTE-0.4)**2)
    SOD(2)=1.12*EPS**C.39

```

```

29C      GO TO 300
      SDD(1)=1./2./EPS/((1.+0.5*EPS)*ALOG(1.3333333/EPS))-0.5*EPS)-1.)
      SDD(2)=-1.23*SQTE*ALOG(0.79*SQTE)
      VGP(2)=0.64*EPS
      SHV(1,ISOLN)=T2
      SHV(2,ISOLN)=EPS
      IF (.NOT.AXISYM.OR.(ISCLN.EQ.2.AND.FSTAG.EQ.0.)) GO TO 440
      IF (ISCLN.EQ.2) GO TO 320
      DD 310 I=1,ISS
      SAVEC(I)=ZCAP(I)
      HCF=0.
      IF (NOTRAN) GO TO 360
      DD 340 I=1,ISS
      NELE=0
      DD 330 J=1,ISC
      IF (LPIJ(I,J).EQ.3) GO TO 330
      IF (LPIJ(I,J).NE.1) GO TO 340
      NELE=NELE+1
      IF (NELE.GT.1) GO TO 340
      CONTINUE
      HCF=HCF+SAVEC(I)*SHJA(I)
      CONTINUE
      HCF=HCF*CRP/ZCM
      CALL TRANSP (TS,PS)
      IF (ERR) RETURN
      HSTAG=CRP/CMA*ZCH
      H=CP*ALL*TMODEL
      HRATIO=HCF/(HSTAG-HW)
      IF (LEWIS.EQ.2) HRATIO=0.
      QSRI=0.C750*SQRT(PS)*(HSTAG-HW)
      0.075=1.8*0.0417. WHERE FACTOR 1.8 CONVERTS ENTHALPIES FROM CAL/GM
      TO BTU/LB
      DD 350 L=1,2
      QS=0.51/(PRW*PR)**0.3*SQ2F(KDIM+1)*SQRT(ZRHO*VISC*U1*VGP(L))*((HSTAG
      1G-HW)
      QFRE(L)=QS*(1.+(FLEWIS**ELN(1))-1.)*HRATIO)
      QFRF(L)=QS*(1.+(FLEWIS**ELN(2))*CATFAC-1.)*HRATIO)
      CKSI=(EPS/(1.+EPS))*2*RI*U1/VISC*30.48
      SCOUT(1)=TS
      SCOUT(2)=CF(1)*ZRHO/RHAP
      SCOUT(3)=PS
      SCOUT(4)=VISC*CF(3)
      SCOUT(5)=FLEWIS
      SCOUT(6)=Z*FN*CRA/ZCM
      SCOUT(7)=7CM
      SCOUT(8)=SIGMA
      SCOUT(9)=PR
      SCOUT(10)=EPS
      IF (NOTRAN) GO TO 370
      SCOUT(11)=QFRE(2)
      SCOUT(12)=QFRF(2)
      SCOUT(13)=QSR1
      SCOUT(14)=SDD(2)
      SCOUT(15)=SDD(1)
      SCOUT(15)=QFRE(1)
      MOD 221
      MOD 222
      MOD 223
      MOD 224
      MOD 225
      MOD 226
      MOD 227
      MOD 228
      MOD 229
      MOD 230
      MOD 231
      MOD 232
      MOD 233
      MOD 234
      MOD 235
      MOD 236
      MOD 237
      MOD 238
      MOD 239
      MOD 240
      MOD 241
      MOD 242
      MOD 243
      MOD 244
      MOD 245
      MOD 246
      MOD 247
      MOD 248
      MOD 249
      MOD 250
      MOD 251
      MOD 252
      MOD 253
      MOD 254
      MOD 255
      MOD 256
      MOD 257
      MOD 258
      MOD 259
      MOD 260
      MOD 261
      MOD 262
      MOD 263
      MOD 264
      MOD 265
      MOD 266
      MOD 267
      MOD 268
      MOD 269
      MOD 270
      MOD 271
      MOD 272
      MOD 273
      MOD 274

```

C C

Line	Code	Statement	Module
275	MOD	SCOUT(17)=QFRF(1,	
276	MOD	SCOUT(18)=CKSI	
277	MOD	SCOUT(19)=HRATIO	
278	MOD	IF (ISOLN.EQ.2) GO TO 390	
279	MOD	HCON=CRF/CMA	
280	MOD	HE=HCON*CH	
281	MOD	HR=HE+SCRT(PR)*HCON*(CHA-CH)	
282	MOD	HSTAR=0.5*(HF+HW)+0.22*(HR-HE)	
283	MOD	TSTAR=CTAP*HSTAR/HS/HCON	
284	MOD	RHOSTR=RHC*RHAP/TSTAR*CTAP*CT	
285	MOD	CALL TRANSP (TSTAR, PRES*PRESA)	
286	MOD	IF (ERR) RETURN	
287	MOD	SCOUT(20)=0.222/PR**0.667*SCUT(RHOSTR*VISC*U1)*{(HR-HW)	
288	MOD	GO TO 390	
289	MOD	IF (.NOT.DATAP) GO TO 390	
290	MOD	DO 380 N=1,20	
291	MOD	SCOUT(N)=0.	
292	MOD	WRITE (6,510) (TITLE2(I,ISOLN),I=1,3)	
293	MOD	LMSC=LIMSCD(ISOLN)	
294	MOD	IF (NOTRAN) LMSC=10	
295	MOD	WRITE (6,520) (QUANT(I,ISOLN),EQUALS,SCOUT(I),I=1,LMSC)	
296	MOD	IF (ISOLN.EQ.2.OR.ISW6B.EQ.0) GO TO 420	
297	MOD	WRITE (6,530) (TITLE2(I,ISOLN),I=1,3)	
298	MOD	IF (IC.EQ.0) GO TO 400	
299	MOD	EMF=SAVEC(I)	
300	MOD	SAVEC(I)=EMF*ZRHO/RHAP*TB7/ZCM	
301	MOD	ISS5PI=(ISS-1)/5+1	
302	MOD	DO 410 K=1,ISS5PI	
303	MOD	LIM1=5*(K-1)+1	
304	MOD	LIM2=MINO(5*K,ISS)	
305	MOD	WRITE (6,540) (HP(I),SAVEC(I),I=LIM1,LIM2)	
306	MOD	IF (IC.EQ.0) GO TO 420	
307	MOD	SAVEC(I)=EMF	
308	MOD	IF (.NOT.DATAP.OR.ISOLN.NE.1) GO TO 430	
309	MOD	NRCOUT=NRCOUT+1	
310	MOD	IMP=IMP+1	
311	MOD	ITYPER=3	
312	MOD	WRITE (ITPOUT) ITPER,IMP,SCOUT(3),SCOUT(11),SCOUT(16),SCOUT(19),SCOUT(19),SCOUT(5),SCOUT(9),SCOUT(4),SCOUT(2)	
313	MOD	IF (FSTAG.EQ.C.AND..NOT.WEDGEM) GO TO 460	
314	MOD	CONTINUE	
315	MOD	GO TO 460	
316	MOD	WRITE (6,550)	
317	MOD	WRITE (6,MODDMP)	
318	MOD	CT=CTSAVE	
319	MOD	IF (NOTRAN) GO TO 470	
320	MOD	IF (WEDGEM) CALL WEDGE	
321	MOD	CALL ELTIME (ET,U)	
322	MOD	DET=ET-ETO	
323	MOD	WRITE (6,560) DET	
324	MOD	RETURN	
325	MOD		
326	MOD		
327	MOD		
328	MOD		
329	MOD	FORMAT (64HOMODEL PARAMETER ROUTINE CALLED FOR A MACH NUMBER LESS	
330	MOD		


```

60  NIT=0
    M2=ISC+2
    ZP=ALOG(PRES)
    DO 90 J=IM,ISC
    AA(J,M2)=QM(J)-CAPX(J)
    IF (NODEPS) GO TO 90
    DO 80 I=1,ISMCNR
    AA(J,M2)=AA(J,M2)-(CDIJ(I,J)-QM(J)*BET(I))*SKIL(I)
    CONTINUE
    AA(M1,M2)=-ZP-ZPA
    DO 100 L=IM,ISSNR
    AA(M1,M2)=AA(M1,M2)+CAPX(L)*(PGJ(L)-CGMU(L))
    DO 150 J=IM,ISC
    DO 150 K=IM,ISC
    IF (J-K) 120,110,120
    AA(J,K)=CAPX(J)
    GO TO 130
    110  AA(J,K)=0.0
    IF (NODEPS) GO TO 150
    DO 140 I=1,ISMCNR
    AA(J,K)=AA(J,K)+CDIJ(I,K)*SKIL(I)*(CDIJ(I,J)-QM(J)*BET(I))
    CONTINUE
    DO 170 K=IM,ISC
    AA(M1,K)=(1.-PGJ(K)+CGMU(K))*CAPX(K)
    IF (NODEPS) GO TO 170
    DO 160 I=1,ISMCNR
    L=I+ISC
    AA(M1,K)=AA(M1,K)+CDIJ(I,K)*(1.-PGJ(L)+CGMU(L))*CAPX(L)
    CONTINUE
    AA(M1,M1)=1.0
    IF (NODEPS) GO TO 190
    DO 180 I=1,ISMCNR
    L=I+ISC
    AA(M1,M1)=AA(M1,M1)+BET(I)*(1.-PGJ(L)+CGMU(L))*CAPX(L)
    DO 210 J=IM,ISC
    AA(J,M1)=0.0
    IF (NODEPS) GO TO 210
    DO 200 I=1,ISMCNR
    L=I+ISC
    AA(J,M1)=AA(J,M1)+BET(I)*CAPX(L)*(CDIJ(I,J)-QM(J)*BET(I))
    CONTINUE
    IF (JJK) 240,220,240
    IF (CAPX(1)-1.0D-20) 230,230,270
    JJK=1
    ISMCNR=ISMCNR-IC
    ISSNR=ISSNR-IC
    NODEPS=ISMCNR.EQ.0
    IM=IM+1
    AA(1,1)=1.0
    DO 250 J=2,M2
    AA(1,J)=0.0
    CAPX(1)=0.0
    LL=ISSNR+1
    DO 260 I=LL,ISS
    CAPX(I)=0.0

```

```

NEW 56
NEW 57
NEW 58
NEW 59
NEW 60
NEW 61
NEW 62
NEW 63
NEW 64
NEW 65
NEW 66
NEW 67
NEW 68
NEW 69
NEW 70
NEW 71
NEW 72
NEW 73
NEW 74
NEW 75
NEW 76
NEW 77
NEW 78
NEW 79
NEW 80
NEW 81
NEW 82
NEW 83
NEW 84
NEW 85
NEW 86
NEW 87
NEW 88
NEW 89
NEW 90
NEW 91
NEW 92
NEW 93
NEW 94
NEW 95
NEW 96
NEW 97
NEW 98
NEW 99
NEW 100
NEW 101
NEW 102
NEW 103
NEW 104
NEW 105
NEW 106
NEW 107
NEW 108
NEW 109
NEW 110

```

```

21. CALL DMSOL (AA,M1,22)
IF (ERR) RETURN
NIT=NIT+1
DO 340 K=IM,ISC
  ZB=1.+AA(K,M2)
  IF (ZB) 280,280,310
  CAPX(K)=CAPX(K)/2.
  IF (CAPX(K)) 290,300,290
  CGMU(K)=ALOG(CAPX(K))
  GO TO 340
  WRITE (6,480) K
  CGMU(K)=CGMU(K)-0.69314718
  GO TO 340
  CAPX(K)=CAPX(K)*ZB
  IF (CAPX(K)) 320,330,320
  CGMU(K)=ALOG(CAPX(K))
  GO TO 340
  WRITE (6,480) K
  CGMU(K)=CGMU(K)+ALOG(ZB)
  CONTINUE
  ZC=1.+AA(M1,M2)
  IF (ZC) 350,350,360
  PRES=PRES/ZC
  IF (PRES) 370,380,370
  PRES=PRES*ZC
  IF (PRES) 370,390,370
  ZP=ALOG(PRES)
  GO TO 400
  WRITE (6,490)
  ZP=ZP-0.69314718
  GO TO 400
  WRITE (6,490)
  ZP=ZP+ALOG(ZC)
  IF (NODEPS) GO TO 430
  DO 420 I=1,ISMCNR
  SKIL(I)=CHI(I)+BET(I)*(ZP+ZPA)
  DO 410 L=IM,ISC
  SKIL(I)=SKIL(I)+CDIJ(I,L)*CGMU(L)
  J=I+ISC
  CGMU(J)=SKIL(I)
  SKIL(I)=EXP(SKIL(I))
  CAPX(J)=SKIL(I)
  CONTINUE
  DO 450 K=IM,M1
  IF (ABS(AA(K,M2)).LE.TEST) GO TO 450
  IF (NIT-NTEST) 70,70,440
  WRITE (6,500)
  CALL DUMP (RNAME)
  RETURN
  CONTINUE
  CH=0.0
  CM=0.0
  DO 460 I=IM,ISSNR
  CM=CM+CAPX(I)*CGI(I)
  CH=CH+CAPX(I)*SHJ(I)

```

```

NEW 111
NEW 112
NEW 113
NEW 114
NEW 115
NEW 116
NEW 117
NEW 118
NEW 119
NEW 120
NEW 121
NEW 122
NEW 123
NEW 124
NEW 125
NEW 126
NEW 127
NEW 128
NEW 129
NEW 130
NEW 131
NEW 132
NEW 133
NEW 134
NEW 135
NEW 136
NEW 137
NEW 138
NEW 139
NEW 140
NEW 141
NEW 142
NEW 143
NEW 144
NEW 145
NEW 146
NEW 147
NEW 148
NEW 149
NEW 150
NEW 151
NEW 152
NEW 153
NEW 154
NEW 155
NEW 156
NEW 157
NEW 158
NEW 159
NEW 160
NEW 161
NEW 162
NEW 163
NEW 164
NEW 165

```

```

RHOBAR=PRES*CM/(CMA*CT)
RHO=(RHOBAR*RUBARA)/(1.+RHOBAR*ROBARP*BZERO)
CH=CH*(CMA/CM)
CH=CH+(PRES/RHOBAR)*(RHOBAR*(ROBARA/RHO)-1.)
DO 470 I=1M,ISSNR
  GJ(I)=CAPX(I)/CM
  SU2=2.*(CHA-CH)
  SU=SQRT(SU2)
  FLUX=RHO*SU
  RETURN
  FORMAT (16H IN NEWRAP,CAPX(.12,3H)=0)
  FORMAT (14H IN NEWRAP,P=0)
  FORMAT (35H100 MANY NEWTON-RAPHSON ITERATIONS)
  END

```

470

C C C

480
490
500

```

NEW 166
NEW 167
NEW 168
NEW 169
NEW 170
NEW 171
NEW 172
NEW 173
NEW 174
NEW 175
NEW 176
NEW 177
NEW 178
NEW 179
NEW 180
NEW 181
NEW 182-

```

1 NEX
2 NEX
3 NEX
4 NEX
5 NEX
6 NEX
7 NEX
8 NEX
9 NEX
10 NEX
11 NEX
12 NEX
13 NEX
14 NEX
15 NEX
16 NEX
17 NEX
18 NEX
19 NEX
20 NEX
21 NEX
22 NEX
23 NEX
24 NEX
25 NEX
26 NEX
27 NEX
28 NEX

```

SUBROUTINE NEXTMP (ITS,XMOD1,XMODEL)
LOGICAL ERR
COMMON /ERROR/ ERR
COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),
1 NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL
COMMON /NEWMP/ FACMP,NMODPT
COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL
COMMON /MODPAR/ XMP1,DXMP,FSTAG, CATFAC,TMODEL,XMODP1,DXMODP,
1 TPLATE,KDIM
DATA ONE /1./
IF (NPRFLS.EQ.1) GO TO 20
IF (.TS.LE.NTS) GO TO 10
XMODEL=1.E30
GO TO 50
10 CALL FINDXC (TSAR(ITS),MBL,XMOD2)
IF (ERR) RETURN
GO TO 40
20 IF (ITS.GT.NTS) GO TO 30
CALL FINDX (TSAR(ITS),ONE,XMOD2)
IF (ERR) RETURN
IF (XMOD1-XMOD2) 30,30,40
XMODEL=XMOD1
XMOD1=XMOD1*FACMP
GO TO 50
40 XMODEL=XMOD2
ITS=ITS+1
RETURN
50 END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE NONEQ
REAL ACOM(30),ELMENT(10),HP(20)
LOGICAL ERR,MODLPT,XMSET,FAILED,COUPLD,AXISYM,WEDGEM,NOREAC,DATAPEN
DOUBLE PRECISION AA,AAA,CJ,GJ,CAPX,DCI,CI,GSQ,DCA,AIN,CDIJ,ONE
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFCH,AMACH,AR,ARBA,ARBB,BZERO,
1 C,CRB,CRS,CSTA,CH,CT,CTAP,CM,CMA,CRA,CRP,
2 CTP,CTPL,CTT,CT,CTB,CTC,CTMAX,CTMX,
3 DELT2,DELTAX,CLOGA,DLOGR,DT,CXB,CXMAX,DATEST,OBTEST,DELT1,
5 SPTTEST,PRESA,PRESB,PRESH,PRHO,RHAP,RHO,PCT,
6 RHOBAR,RHOC,RHOP,RHPL,RHTR,ROBARA,ROBARP,SCPG,SDT,
7 SEN,SHPG,SC,SL,SL64,SM,SU,SUMG,
8 TEST,TESTB,TPRINT,TSTOP,TUP,ZP,ZPA
COMMON BE(64),BET(20),BLBK(31),CAI(64),CAPXTH(20),
1 CCPJ(20),CEACT(64),CGI(20),CGMU(20),CHI(64),CHI(20),
2 CLNIMC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),GJA(20),
3 GJB(20),PERTGJ(20),PGJ(20),PI(64),PICH(64),QM(20),
4 GO(64),SAJ(20),SDCH(64),SENT(20),SHJ(20),SHJA(20),
5 SKIL(20),SS(20),TFA(20),TFB(20),TFC(20),
6 TFD(20),TFE(20),TFK(20),THEV(20),XMJAT(20),XNUI(64)
COMMON BETA(64,20),ELJ(10,20),GELJ(10,20),
1 XNUIJ(64,20),XNUJJP(64,20)
COMMON IC,IM,INEQ,INEQV,IP,IRUN,ISC,ISCP1,
1 ISMC,ISMCMR,ISR,ISS,ISSN,ISSP1,ISSP2,ISSP3,ISSP4,
2 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,ISW5A,
3 ISW5B,ISW6A,ISW6B,IUPD,IZERO,JJK,LC,M1,NFIT,
4 NIT,NNN,NNS,NQS,NOT,NTEST
COMMON IGM(20),ITB(5),KUR(64,20),LPIJ(20,10)
COMMON ACOM,ELMENT,HP
DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBN
1J(20),SDGJ(20),SHJAP(20),THEVP(20)
EQUIVALENCE (AAA(1,1),AAA(1,1)),(BTA(1,1),BTA(1,1)),
1 (BLRK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
DIMENSION CI(10),DCI(10),DCA(10),GSQ(20)
EQUIVALENCE (GSQ(1),AA(1,24))
EQUIVALENCE (CCI(1),AA(1,23))
COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAP
COMMON /INGNE/ AIN(10,10)
COMMON /BL/ DEBEL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
1 SW,RO,JDIM,IPCINT
COMMON /NEQ/ OMDST(2),DDELBL(2)
COMMON /MODPT/ TSDIAM(20),TSAR(20),NTS,MBL
COMMON /ERROR/ ERR
COMMON /CERROR/ FAILED
COMMON /BLNE/ XI(2),ORDINP(2),DELBLP(2),DOBLP(2),AMPP,AMP,
1 COUPLD,ISMD
COMMON /HLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),
1 TAUW(2)
DIMENSION BLNEAR(10),ISUPR(64),CAISV(64)
EQUIVALENCE (BLNEAR(1),XI(1)),(ISUPR(1),AA(1,1))
COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DHLL,DCHRRAT,CCHI,TTEST,
1 GTEST

```


ORIGINAL PAGE IS
OF POOR QUALITY

-480-

```
SCD=2,
ITB(1)=0
NNS=0
NNN=0
ISSP1=ISS+1
ISSP2=ISS+2
ISSP3=ISS+3
ISSP4=ISS+4
INEQ=0
DO 30 I=1,ISR
PI(I)=1.0
ZZZ=CAI(I)
CAI(I)=ALOG(ZZZ)-ETAI(I)*ALOGT
CAISV(I)=CAI(I)
BE(I)=0.0
XNUI(I)=0.0
DO 30 J=1,ISS
BETA(I,J)=XNUIJP(I,J)-XNUIJ(I,J)
BE(I)=BE(I)+BETA(I,J)
XNUI(I)=XNUI(I)+XNUIJ(I,J)
ZZZ=CMA/CRP
CSTA=5*ALOG(ZZZ)
SL64=SL/6468.
COMPUTE AND TEST RANK OF BETA MATRIX
IX=0
I=0
I2=1
I=I+1
IF (I-ISS) 50,50,150
DO 70 J=12,ISR
IF (BETA(J,I)) 60,70,60
KA=J
IX=IX+1
GO TO 80
CONTINUE
GO TO 40
K=KA
DO 90 M=1,ISS
AB=BETA(K,M)
BETA(K-M)=BETA(IX,M)
BETA(IX)=AB
I2=IX
IF (I2-ISS) 100,100,150
DO 120 N=12,ISR
CA=BETA(I,I)/BETA(IX,I)
IF (CA) 110,140,110
DO 130 N=1,ISS
BETA(M,N)=BETA(M,N)-CA*BETA(IX,N)
IF (ABS(BETA(M,N))-0.01) 120,120,130
BETA(1,N)=0.0
CONTINUE
GO TO 40
IF (IX-ISM) 160,170,160
```

```
NON 111
NON 112
NON 113
NON 114
NON 115
NON 116
NON 117
NON 118
NON 119
NON 120
NON 121
NON 122
NON 123
NON 124
NON 125
NON 126
NON 127
NON 128
NON 129
NON 130
NON 131
NON 132
NON 133
NON 134
NON 135
NON 136
NON 137
NON 138
NON 139
NON 140
NON 141
NON 142
NON 143
NON 144
NON 145
NON 146
NON 147
NON 148
NON 149
NON 150
NON 151
NON 152
NON 153
NON 154
NON 155
NON 156
NON 157
NON 158
NON 159
NON 160
NON 161
NON 162
NON 163
NON 164
NON 165
```

```

160 WRITE (6,1090)
161 C ERROR EXIT NO. 1 *****
162 C BETA MATRIX OF INCORRECT RANK *****
163 C IERR=1 *****
164 C GO TO 1000 *****
165 C RESET BETA MATRIX (DESTROYED DURING RANK CALCULATION) *****
166 C DO 180 I=1,ISR *****
167 C DO 180 J=1,ISS *****
168 C BETA(I,J)=XNUIJP(I,J)-XNUIJ(J,I) *****
169 C AR=C.0 *****
170 C LC=0 *****
171 C DELT=DELT1 *****
172 C DO 200 I=1,ISC *****
173 C CI(I)=0. *****
174 C DO 190 J=1,ISS *****
175 C C(I)=CI(I)+GJA(J)*LPIJ(J,I) *****
176 C CONTINUE *****
177 C GO TO 220 *****
178 C END OF INITIALIZATIONS *****
179 C IF (MODLPT) CALL NEXTMP (ITS,XMOD1,XMODEL) *****
180 C GO TO 230 *****
181 C CALL DERIVS *****
182 C IF (IERR) RETURN *****
183 C IF (.NOT. FAILED) GO TO 230 *****
184 C ERROR EXIT NO. 2 *****
185 C IERR=2 *****
186 C GO TO 1000 *****
187 C IF (XMSSET.OR. AMACH.LT.1.5) GO TO 240 *****
188 C XMSSET=.TRUE. *****
189 C XMOD1=XMPI *****
190 C ITS=1 *****
191 C CALL NEXTMP (ITS,XMOD1,XMODEL) *****
192 C IF (IUPD) 250,250,250 *****
193 C TESTS FOR SWITCH TO DOWNSTREAM REGION *****
194 C IF (OLOGA) 450,260,260 *****
195 C IF (CX) 450,450,270 *****
196 C IF (AR) 300,280,300 *****
197 C AR=1.0 *****
198 C DO 290 I=1,NPRELS *****
199 C OMDST(I)=1.-DELBL(I) *****
200 C IF (AFNX.LT. DATEST) GO TO 450 *****
201 C SAVE DATA AT SWITCH POINT FOR POSSIBLE RESTART *****
202 C TB(1)=CX *****
203 C TB(2)=CT *****
204 C TB(3)=SC *****
205 C TB(4)=CTT *****
206 C TB(5)=DELTA *****
207 C TB(6)=HDELX *****
208 C TB(7)=CTE *****
209 C TB(8)=SUD *****
210 C TB(9)=CHA *****
211 C ITB(1)=INEQ *****
212 C ITB(2)=NNS *****
213 C ITB(3)=NNN *****
214 C ITB(4)=NRROUT *****
215 C *****
216 C *****
217 C *****
218 C *****
219 C *****
220 C *****
221 C *****
222 C *****
223 C *****
224 C *****
225 C *****
226 C *****
227 C *****
228 C *****
229 C *****
230 C *****
231 C *****
232 C *****
233 C *****
234 C *****
235 C *****
236 C *****
237 C *****
238 C *****
239 C *****
240 C *****
241 C *****
242 C *****
243 C *****
244 C *****
245 C *****
246 C *****
247 C *****
248 C *****
249 C *****
250 C *****
251 C *****
252 C *****
253 C *****
254 C *****
255 C *****
256 C *****
257 C *****
258 C *****
259 C *****
260 C *****
261 C *****
262 C *****
263 C *****
264 C *****
265 C *****
266 C *****
267 C *****
268 C *****
269 C *****
270 C *****
271 C *****
272 C *****
273 C *****
274 C *****
275 C *****
276 C *****
277 C *****
278 C *****
279 C *****
280 C *****
281 C *****
282 C *****
283 C *****
284 C *****
285 C *****
286 C *****
287 C *****
288 C *****
289 C *****
290 C *****
291 C *****
292 C *****
293 C *****
294 C *****
295 C *****
296 C *****
297 C *****
298 C *****
299 C *****
300 C *****
301 C *****
302 C *****
303 C *****
304 C *****
305 C *****
306 C *****
307 C *****
308 C *****
309 C *****
310 C *****
311 C *****
312 C *****
313 C *****
314 C *****
315 C *****
316 C *****
317 C *****
318 C *****
319 C *****
320 C *****
321 C *****
322 C *****
323 C *****
324 C *****
325 C *****
326 C *****
327 C *****
328 C *****
329 C *****
330 C *****
331 C *****
332 C *****
333 C *****
334 C *****
335 C *****
336 C *****
337 C *****
338 C *****
339 C *****
340 C *****
341 C *****
342 C *****
343 C *****
344 C *****
345 C *****
346 C *****
347 C *****
348 C *****
349 C *****
350 C *****
351 C *****
352 C *****
353 C *****
354 C *****
355 C *****
356 C *****
357 C *****
358 C *****
359 C *****
360 C *****
361 C *****
362 C *****
363 C *****
364 C *****
365 C *****
366 C *****
367 C *****
368 C *****
369 C *****
370 C *****
371 C *****
372 C *****
373 C *****
374 C *****
375 C *****
376 C *****
377 C *****
378 C *****
379 C *****
380 C *****
381 C *****
382 C *****
383 C *****
384 C *****
385 C *****
386 C *****
387 C *****
388 C *****
389 C *****
390 C *****
391 C *****
392 C *****
393 C *****
394 C *****
395 C *****
396 C *****
397 C *****
398 C *****
399 C *****
400 C *****
401 C *****
402 C *****
403 C *****
404 C *****
405 C *****
406 C *****
407 C *****
408 C *****
409 C *****
410 C *****
411 C *****
412 C *****
413 C *****
414 C *****
415 C *****
416 C *****
417 C *****
418 C *****
419 C *****
420 C *****
421 C *****
422 C *****
423 C *****
424 C *****
425 C *****
426 C *****
427 C *****
428 C *****
429 C *****
430 C *****
431 C *****
432 C *****
433 C *****
434 C *****
435 C *****
436 C *****
437 C *****
438 C *****
439 C *****
440 C *****
441 C *****
442 C *****
443 C *****
444 C *****
445 C *****
446 C *****
447 C *****
448 C *****
449 C *****
450 C *****
451 C *****
452 C *****
453 C *****
454 C *****
455 C *****
456 C *****
457 C *****
458 C *****
459 C *****
460 C *****
461 C *****
462 C *****
463 C *****
464 C *****
465 C *****
466 C *****
467 C *****
468 C *****
469 C *****
470 C *****
471 C *****
472 C *****
473 C *****
474 C *****
475 C *****
476 C *****
477 C *****
478 C *****
479 C *****
480 C *****
481 C *****
482 C *****
483 C *****
484 C *****
485 C *****
486 C *****
487 C *****
488 C *****
489 C *****
490 C *****
491 C *****
492 C *****
493 C *****
494 C *****
495 C *****
496 C *****
497 C *****
498 C *****
499 C *****
500 C *****

```

```

310 DO 310 I=1,ISS
    TB(I+10)=GJ(I)
    BLBK(I)=RSA
    IF (ISW3B.EQ.0) GO TO 340
    DO 320 I=1,10
    BLBK(I+1)=BLNEAR(I)
    BLBK(I+2)=XPB
    BLBK(I+3)=DXPB
    DO 330 I=1,NPRFLS
    BLBK(I+13)=BLINT(I)
    BLBK(I+15)=XSN(I)
    BLBK(I+17)=DELBL(I)
    BLBK(I+19)=DDELBL(I)
    SWITCH TO DOWNSTREAM REGION
    CALL THROAT
    IF (ERR) RETURN
    GO TO 450
    IF (DLOGR) 450,450,360
    WRITE (6,1100)
    IF ((AFNX-DATEST)/DATEST.LT.0.05) GO TO 380
    ERROR EXIT NO. 3 *****
    POSITIVE DLOGR FAR BEYOND SWITCH POINT
    IERR=3
    GO TO 1000
    SWITCH TO DOWNSTREAM REGION WAS PREMATURE.
    INCREASE DATEST AND RESTART UPSTREAM SOLUTION AT SWITCH POINT.
    DATEST=2.*(DATEST-1.))+1.
    IUPD=1
    ARBB=ARBB+1.0
    IF (ARBB.LT.ARBA) GO TO 390
    ERROR EXIT NO. 4 *****
    IERR=4
    GO TO 1000
    TSZ=TPRINT
    TPRINT=0.0
    CALL PRTA
    IF (ERR) RETURN
    ITAPE=NRCONT-ITB(4)
    IF (ITAPE.EQ.0) GO TO 410
    BACKSPACE BINARY OUTPUT TAPE TO ELIMINATE RECORDS OF INVALID
    STEPS BEYOND SWITCH POINT
    DO 400 I=1,ITAPE
    BACKSPACE ITPUT
    TPRINT=TSZ
    RESET DATA FOR RESTART AT SWITCH POINT
    CX=TB(1)
    CT=TB(2)
    SC=TB(3)
    CTT=TB(4)
    DELTAX=TB(5)
    HDELX=TB(6)
    CTE=TB(7)
    SCD=TB(8)
    CHA=TB(9)
    INEQ=ITB(1)

```

```

420 NNS=ITB(2)
NNN=ITB(3)
NRCOUT=ITB(4)
DO 420 I=1,ISS
GJ(I)=TB(I+10)
RSA=BLBK(I)
IF (ISW3B.EQ.0) GO TO 220
DO 430 I=1,10
BLNEAR(I)=BLBK(I+1)
XP(I)=BLBK(I2)
DXPB=BLBK(I3)
DO 440 I=1,NPRFLS
BLINT(I)=BLBK(I+13)
XSN(I)=BLBK(I+15)
DELBL(I)=BLBK(I+17)
DDELBL(I)=BLBK(I+19)
ISMD=30
COUPLD=.FALSE.
GO TO 220
CONTINUE
IF (INEQ) 610,460,610
PERTURBATION SOLUTION
CALL PERT
IF (ERR) RETURN
IF (IUPD.EQ.C.AND.DELT.EQ.DELT1) DELT=DELT5V
DETERMINE MAXIMUM AND MINIMUM PERTURBATIONS IN CHI(I)
DCHMAX=C.
DCHMIN=1.E30
DO 480 I=1,ISR
ADCH=ABS(SDCHI(I))
IF (ADCH-DCHMAX) 480,480,470
DCHMAX=ADCH
IMAX=I
DCHMIN=AMINI(CC IN,ADCH)
WRITE (6,1110) DCHMIN,DCHMAX,IMAX
IF (DCHMAX.EQ.0.) GO TO 580
RATIO=DCHMIN/DCHMAX
IF (RATIO.GE.DCHPRAT) GO TO 510
EXCESSIVELY SMALL RATIO OF MINIMUM AND MAXIMUM PERTURBATIONS
IN CHI(I)
DO 500 J=1,ISS
IF (BETA(IMAX,J)*PICH1(IMAX).GE.0.) GO TO 500
IF (GJ(J).GT.GAMIN) GO TO 500
IF (IRA.LE.50) GO TO 490
WRITE (6,1010)
GO TO 530
ARTIFICIAL INCREASE OF RATE CONSTANT TO PREVENT PREMATURE START
OF INTEGRATION
RATIO2=1.1*DCHPRAT/RATIO
CAI(IMAX)=CAI(IMAX)+ALOG(RATIO2)
WRITE (6,1020) RATIO,HP(J),IMAX,GJ(J),IMAX,RATIO2
IRA=IRA+1
GO TO 220
CONTINUE
TESTS FOR SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION
500
C

```

```

510 IF (DCHMAX.LT.CCHI) GO TO 580
C IF (DCHMAX.LE.PCTEST*CCHI) GO TO 540
C MAXIMUM SDCHI(I) IS TOO LARGE -- BACKSTEP THE PERTURBATION
  SOLUTION TO FIND THE CORRECT SWITCHING POINT
  DELT=DELT/2.0
  ISUBD=ISUBD+1
  IF (ISURD.GT.23) GO TO 520
  CT=CT+DELT
  CTE=CT
  NAMELIST /NEQDMP/ INEQ,IUPD,DLOGA,AR,CX,CT,DELT,AFNTS.
1  UP,SDCHI,DDELBL,PCT,PRHO,GJ,PERTGJ
  IF (ISWSB.NE.0) WRITE (6,NEQDMP)
  GO TO 600
  WRITE (6,1120) ISUBC
  WRITE (6,NEQDMP)
  ERROR EXIT NO. 5 *****
  IERR=5
  GO TO 1000
  SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION
  INEQ=1
  DXM=0.01*DCHMIN/DCHLL
  IF (DELTAX.LE.DXM) GO TO 550
  DELTAX=DXM
  HDELX=0.5*DELTAX
  TB(5)=DELTAX
  TB(6)=HDELX
  DXOLD=DELTAX
  CT=CT+PCT
  CTE=CT
  RHQ=RHQ+PRHO
  DO 560 I=1,ISR
  IF (CAI(I).EQ.CAISV(I)) GO TO 560
  CAI(I)=CAISV(I)
  WRITE (6,1030) I
  CONTINUE
  DO 570 J=1,ISS
  GJ(J)=GJ(J)+PERTGJ(J)
  CALL DERIV
  IF (ERR) RETURN
  IF (.NOT.FAILED) GO TO 610
  ERROR EXIT NO. 6 *****
  IERR=6
  GO TO 1000
  CONTINUE PERTURBATION SOLUTION
  IF (CX.LT.XMODEL) GO TO 590
  MODLPT=.TRUE.
  CALL NEXTMP (ITS,XMOD1,XMODEL)
  PRINT POINT IN PERTURBATION SOLUTION
  CALL PRTA
  IF (ERR) RETURN
  IF (CX.GE.CXMAX) GO TO 660
  IF (CT.LE.TSTCP) GO TO 660
  ISUBD=J
  CT=CT-DELT
  CTE=CT

```

```

60 CALL NEWRAP
   IF (ERR) RETURN
   AFNTS=SM/FLUX
   CALL AXFIT
   IF (ERR) RETURN
   GO TO 220
C    NUMERICAL INTEGRATION
510 IF (CT-1.0) 630,630,620
520 WRITE (6,1130)
C    ERROR EXIT NO. 7 *****
   IERR=7
   GO TO 1000
630 IF (CX.GE.CXMAX) GO TO 660
   IF (CT.LE.TSTOP) GO TO 660
   NNN=NNN+1
   PRINT POINT IN SOLUTION BY INTEGRATION
   CALL PRTA
   IF (ERR) RETURN
   IF (NNN-NQS) 670,640,640
   DELTAX=SC*DELTAX
   HDELX=DELTAX/2.0
   NNS=NNS+1
   NNN=0
   IF (NNS-NQS) 670,650,650
   SC=SC+0.1
   SC=SC
   NNS=0
   GO TO 670
C    FINAL POINT
660 TPRINT=0.0
   CALL PRTA
   AXISYME=.TRUE.
   CALL MOCCL
   WRITE (6,1160) ISTEPS
C    MAIN RETURN STATEMENT.....
C    RETURN
C    SAVE DATA AT BEGINNING OF INTEGRATION STEP FOR POSSIBLE RESTART
670 DO 680 I=1,ISS
   GJB(I)=GJ(I)
680 CONTINUE
   CXB=CX
   CTB=CT
   CARB=APNX
   RHOB=RHO
   CRB=SEN
   CHB=CHA
   DELBLB(1)=DELBL(1)
   IF (NPR.S.EQ.2) DELBLB(2)=DELBL(2)
   IF (CX+LELTAX.LT.XWCDEL) GO TO 700
   MOCCLPT=.TRUE.
   DXSAVE=DELTAX
   HDXSAVE=HDELX

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

700 DELTAX=XMODEL-CX
      HDELTAX=0.5*DELTAX
      IF (CX+DELTAX.LT.CXMAX) GO TO 710
      DELTAX=CXMAX-CX
      HDELTAX=0.5*DELTAX
      INTEGRATION STEP
      CALL RNKT RETURN
      IF (ERR) RETURN
      IF (FAILED) GO TO 740
      IF (.NOT.MODLPT) GO TO 720
      DELTAX=DXSAVE
      HDELTAX=HDXSAV
      CONTINUE
720 C TESTS FOR VALID INTEGRATION STEP
      IF FAIL=1
        IF (ABS(SDT/CT).GT.YTEST) GO TO 740
        IF (NT.EQ.1) GO TO 730
      IF FAIL=2
        IF (ABS(SDTE/CTE).GT.YTEST) GO TO 740
      IF FAIL=3
        IF (ABS(SDCHA/CHA).GT.HTEST) GO TO 740
      IF FAIL=4
        IF (LOGA*CX.GE.-0.01) GO TO 760
      IF (LOGA*CX.GE.-0.01) GO TO 760
      INTEGRATION STEP FAILED. REDUCE STEP SIZE AND RESTART AT
      BEGINNING OF STEP
730 C DELTAX=DELTAX/SCD
      HDELTAX=DELTAX/2.0
      MODLPT=.FALSE.
      ICOUNT=ICOUNT+1
      IF (ICOUNT.GT.30.OR.DELTAX.LE.1.E-10) GO TO 990
      IF (YCOUNT) =IFAIL
      DO 750 I=1,ISS
750 C GJ(I)=GJR(I)
      CTE=CTR
      CTE=CTEB
      CX=CX8
      AFNX=CARB
      RHQ=RHQB
      SEN=CRS
      CHA=CHB
      DELBL(1)=DELRBLB(1)
      IF (NFRFLS.EQ.2) DELBL(2)=DELRBLB(2)
      NNN=0
      NNS=0
      SC=SC-0.1
      SC=AMAX1(SC,1.1)
      SCD=1.1*SCD
      CALL DERIVS
      IF (ERR) RETURN
      IF (.NOT.FAILED) GO TO 690
      ERROR EXIT NO. 8 *****
      IF N=8
        GO TO 1000
      FURTHER VALIDITY CHECKS ON STEP
760 C DO 790 I=1,ISS

```

```

770 IFAIL=I+10
      GJ(I)=GJB(I)+SDGJ(I)
      IF (GJ(I).LT.0.) GO TO 770
      IF (ABS(SDGJ(I)/GJB(I)).LE.GTEST) GO TO 790
      IF (GJB(I).GE.GAMIN) GO TO 740
      IF (SDGJ(I).LE.0.) GO TO 740
      SUPPRESS REACTIONS INVOLVING STEP-SIZE CONTROLLING SPECIES
      WITH CONCENTRATION BELOW GAMIN AND FALLING
      NSUPR=0
      DO 780 K=1,ISR
      IF (BETA(K,I).EQ.0.) GO TO 780
      NDREAC(K)=.TRUE.
      NSUPR=NSUPR+1
      ISUPR(NSUPR)=K
      CONTINUE
      WRITE (6,1053) HP(I),GJB(I),(ISUPR(K),K=1,NSUPR)
      GO TO 740
      VALID STEP. APPLY INCREMENTS TO DEPENDENT VARIABLES.
      CONTINUE
      CT=CTB+SDT
      IF (NT.EQ.1) GO TO 800
      CTE=CTEB+SDTE
      CHA=CHB+SDCHA
      RESTORE CONSERVATION OF CHEMICAL ELEMENTS
      DO 820 J=1,ISC
      CJ=0.
      DO 810 I=1,ISS
      CJ=CJ+GJ(I)*LPIJ(I,J)
      DCI(J)=CI(J)-CJ
      DO 830 I=1,ISS
      GSQ(I)=GJ(I)**2
      IF (GSQ(I).GT.9.) OR (I.GT.ISC) GO TO 830
      WRITE (6,1060) I
      ERROR EXIT NO. 9 *****
      IERR=9
      GO TO 1000
      CONTINUE
      ISMCP1=ISMCP+1
      DO 850 K=1,ISC
      DCA(K)=0.
      DO 840 J=1,ISC
      DCA(K)=DCA(K)+DCI(J)*AIN(J,K)
      CONTINUE
      DO 890 I=1,ISMCP
      II=I+ISC
      AA(I,ISMCP1)=0.
      DO 860 K=1,ISC
      AA(I,ISMCP1)=AA(I,ISMCP1)+CDIJ(I,K)*DCA(K)
      AA(I,ISMCP1)=AA(I,ISMCP1)*GSQ(II)
      DO 880 L=1,ISMCP
      AA(I,L)=0.
      DO 870 K=1,ISC
      AA(I,L)=AA(I,L)+CDIJ(I,K)*CDIJ(L,K)/GSQ(K)
      AA(I,L)=AA(I,L)*GSQ(II)

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

880 IF (I.EQ.0) AA(I,L)=AA(I,L)+ONE
890 CONTINUE
    CALL DSM SOL (AA, ISMC, 22)
    IF (ERR) RETURN
    DO 900 I=1, ISMC
      II=I+ISC
      GJ(II)=GJ(II)+AA(I, ISMCP1)
      DO 930 K=1, ISC
        GJ(K)=0.
        DO 910 J=1, ISC
          GJ(K)=GJ(K)+CI(J)*AIN(J,K)
        DO 920 L=1, ISMC
          I=L+ISC
          GJ(K)=GJ(K)-GJ(I)*CDIJ(L,K)
        CONTINUE
        DO 940 I=1, ISS
          IFAIL=I+30
          IF (GJ(I)) 740, 940, 940
        CONTINUE
        COMPUTE SUPPLEMENTARY FLOW VARIABLES AND DERIVATIVES AT END OF
        STEP (AND BEGINNING OF NEXT STEP)
        CALL DERIVS
        IF (ERR) RETURN
        IFAIL=60
        IF (IFAIL) GO TO 740
        IF (NT.EQ.1) GO TO 960
        FINAL VALIDITY CHECK IN ELECTRONIC NONEQUILIBRIUM MODEL --
        TEST ON CHANGE IN ENERGY TRANSFER TO ELECTRONS
        IF (QECLD.EQ.0.) GO TO 950
        IFAIL=5
        DQEM=AMAX1(QTEST*ABS(QEOLD), DQMAX)
        IF (ABS(QDPE-QEOLD).GT.DQEM) GO TO 740
        DQMAX=AMAX1(DQMAX, DQEM)
        QEOLD=GDPE
        FULLY SUCCESSFUL STEP. SET UP FOR NEXT STEP.
        SCD=SC
        TPE=CT*CTAP
        TEP=CTE*CTAP
        IF (IS5.GE.0) GO TO 970
        WRITE (6,1140) CX, DELTAX, TP, TEP, CHA, QDPE, ICOUNT
        IF (ICOUNT.EQ.0) GO TO 980
        WRITE (6,1040) (IF(K), K=1, ICOUNT)
        ICOUNT=0
        ISTEPS=ISTEPS+1
        DELTAX=AMAX1(DELTAX, 0.7*DXOLD)
        DXOLD=DELTAX
        GO TO 210
        WRITE (6,1150)
        ERROR EXIT NO. 10*****
        IERR=10
        IF (IUPD.EQ.0) GO TO 370
        WRITE (6,1070) IERR
        CALL DUMP (RNAME)
        RETURN
1000

```

```

NON 551
NON 552
NON 553
NON 554
NON 555
NON 556
NON 557
NON 558
NON 559
NON 560
NON 561
NON 562
NON 563
NON 564
NON 565
NON 566
NON 567
NON 568
NON 569
NON 570
NON 571
NON 572
NON 573
NON 574
NON 575
NON 576
NON 577
NON 578
NON 579
NON 580
NON 581
NON 582
NON 583
NON 584
NON 585
NON 586
NON 587
NON 588
NON 589
NON 590
NON 591
NON 592
NON 593
NON 594
NON 595
NON 596
NON 597
NON 598
NON 599
NON 600
NON 601
NON 602
NON 603
NON 604
NON 605

```



```

SUBROUTINE NRMAX
LOGICAL ERR
REAL ACOM(30),ELMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFNCH,AMACH,
C,CARB,RS,CSTA,CH,CT,CTAP,
CTP,C,L,DLOGA,DLOGR,PRESTH,PRHO,RHAP,
SPCTEST,PRES,PRESA,RHOP,RHPL,RHTR,ROBARA,ROBARP,
GRHUPAR,RHOC,RHOP,SC,SL,SL64,
SEN,SHPG,SC,SL,SL64,
TEST,TESTB,TPRINT,TSTOP,
COMMON BE(1),
CCPJ(20),CEACT(64),
CLNPI(64),
GJBJ(20),PERTGJ(20),
GQ(64),SAJ(20),SDCHI(64),
SKIL(20),SS(20),TB(30),
TFD(20),TFE(20),
COMMON BETA(64,20),ELJ(10,20),
XNUIJ(64,20),XNUJJP(64,20)
COMMON IC,IM,INEQ,INEQV,ISSP1,ISSP2,ISSP3,ISSP4,
ISMC,ISMCNR,ISR,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,
ISW1A,ISW1B,ISW2A,ISW2B,IUPD,IZERO,JJK,
ISW5D,ISW5A,ISW6R,ISW6R,NQS,
NIT,NNN,NNS,NQT,NTEST
COMMON IGJ(20),IGM(20),ITB(5),KUR(64,20),LPIJ(20,10)
COMMON ACOM,ELMENT,HP
DIMENSION AAA(22,24),BTAA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBNRM
1J(20),SDGJ(20),SHJAP(20),THEVP(20)
EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTAA(1,1)),
(ELBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
1 (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),
2 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
3 COMMON /ERROR/ ERR
M1=ISC+1
DELT=DELT1
PRES=1.0
TI=1.0
CT=1
DO 10 J=1,ISS
GJ(J)=GJA(J)
F1=0.
CT=CT-DELT
T2=CT
CALL NEWRAP
IF (ERR) RETURN
F2=FLUX
CT=CT-DELT
T3=CT
CALL NEWRAP
IF (ERR) RETURN
F3=FLUX

```

ORIGINAL PAGE IS
OF POOR QUALITY

10

20

30

SUBROUTINE OUT
 WRITE (6,10)
 WRITE (6,20)
 WRITE (6,30)
 WRITE (6,40)
 WRITE (6,50)
 WRITE (6,60)
 WRITE (6,70)
 WRITE (6,80)
 WRITE (6,90)
 WRITE (6,100)
 WRITE (6,110)
 RETURN

ORIGINAL PAGE IS
 OF POOR QUALITY

FORMAT (23H1DEFINITIONS OF SYMBOLS//)

FORMAT (31H ARAT = GEOMETRIC AREA RATIO/31H ARATEF = EFFECTIVE
 1 AREA RATIO/96H DELREF = EQUILIBRIUM SHOCK STANDOFF DISTANCE ON FOUT
 2 LAT-FACED MODEL, DIVIDED BY MODEL NOSE RADIUS/99H DELREH = EQUILIBRIUM
 3 BRIUM SHOCK STANDOFF DISTANCE ON HEMISPHERICAL MODEL, DIVIDED BY MOUT
 4 MODEL NOSE RADIUS/91H DELREF = FROZEN SHOCK STANDOFF DISTANCE ON FOUT
 5 SLAT-FACED MODEL, DIVIDED BY MODEL NOSE RADIUS/94H DELRFH = FROZEN
 6 SHOCK STANDOFF DISTANCE ON HEMISPHERICAL MODEL, DIVIDED BY MODEL
 7 NOSE RADIUS/73H DELSTR = DISPLACEMENT THICKNESS OF BOUNDARY LAYER
 8 ON NOZZLE WALL (INCH)
 9 FORMAT (64H DELSTW = BOUNDARY LAYER DISPLACEMENT THICKNESS ON WEDOUT
 1GE (INCH)/35H DIAM = NOZZLE DIAMETER (INCHES)/56H EPSLE = DENOUT
 2SITY RATIO ACROSS EQUILIBRIUM NORMAL SHOCK/51H EPSLF = DENSITY RATIO
 3ATIO ACROSS FROZEN NORMAL SHOCK/52H GAMMA = FROZEN SPECIFIC-HEAT RATIO
 4 RATIO ACROSS FROZEN STREAM/43H H = ENTHALPY IN FROZEN STREAM (BTU/LOUT
 5H)/58H HEIGHT = SECOND TRANSVERSE DIMENSION OF CHANNEL (INCHES)/30UT
 67H HR = RECOVERY ENTHALPY (BTU/LB)/74H HRATE = RATIO OF DISOUT
 7SOCATION ENTHALPY TO (HS-HW) FOR EQUILIBRIUM SHOCK)
 8 FORMAT (69H HRATE = RATIO OF DISSOCIATION ENTHALPY TO (HS-HW) FOUT
 9R FROZEN SHOCK/39H HS = STAGNATION ENTHALPY (BTU/LB)/84H K2PFOUT
 2E = SHOCK-BOUNDARY LAYER INTERACTION PARAMETER PER FOOT FOR EQUILIBRIUM
 3LIBRIUM SHOCK/79H K2PFF = SHOCK-BOUNDARY LAYER INTERACTION PARAMETER
 4ETER PER FOOT FOR FROZEN SHOCK/83H LEE = ATOM-MOLECULE LEWIS NOUT
 5UMBER AT STAGNATION CONDITION FOR EQUILIBRIUM SHOCK/78H LEF = OUT
 6ATM-MOLECULE LEWIS NUMBER AT STAGNATION CONDITION FOR FROZEN SHOCK
 7K/37H M = MACH NUMBER IN FREE STREAM/47H MU = VISCOSITYOUT
 8 IN FREE STREAM (LB/FT-SEC)/78H MUT2E = VISCOSITY AT STAGNATION OUT
 9CONDITION FOR EQUILIBRIUM SHOCK (LB/FT-SEC)/73H MUT2F = VISCOSITOUT
 0Y AT STAGNATION CONDITION FOR FROZEN SHOCK (LB/FT-SEC)/52H MW OUT
 1 = MOLECULAR WEIGHT IN FREE STREAM (GM/MOLE)
 2 FORMAT (83H MW12E = MOLECULAR WEIGHT AT STAGNATION CONDITION FOROUT
 3 EQUILIBRIUM SHOCK (GM/MOLE)/78H MW12F = MOLECULAR WEIGHT AT STAGNATION
 4GNATION CONDITION FOR FROZEN SHOCK (GM/MOLE)/53H N = BOUNDAROUT
 5Y-LAYER PRESSURE-GRADIENT PARAMETER/40H P = PRESSURE IN FREEOUT
 6 STREAM (ATM)/71H PRE = PRANDTL NUMBER AT STAGNATION CONDITIONOUT
 7 FOR EQUILIBRIUM SHOCK/66H PRF = PRANDTL NUMBER AT STAGNATION OUT
 8CONDITION FOR FROZEN SHOCK/68H PRREF = PRANDTL NUMBER AT REFERENOUT
 9CE TEMPERATURE IN BOUNDARY LAYER/58H PT2E = STAGNATION PRESSUREOUT
 0 FOR EQUILIBRIUM SHOCK (ATM)/53H PT2F = STAGNATION PRESSURE FOROUT

9 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 56
 9L (ATM)) = STAGNATION-POINT HEAT FLUX TIMES SORT(R) FOUT OUT 57
 10 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 58
 11 EQUILIBRIUM SHOCK ON A LAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDO 59
 12 EQUILIBRIUM SHOCK ON A LAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDO 60
 2DARY LAYER/20X.40H(BTU/SQ FT-SEC)FT*0.5 FAY-RIDDELL/124H QEOUT 61
 3FF = STAGNATION-POINT HEAT FLUX TIMES SORT(R) FOR EQUILIBRIUM SOUT 62
 4HOCK ON A FLAT-FACED MODEL WITH A FROZEN BOUNDARY LAYER/20X.40H(BTOUT 63
 5U/SQ FT-SEC)FT*0.5 FAY-RIDDELL/127H QEHE = STAGNATION-POOUT 64
 6INT HEAT FLUX TIMES SORT(R) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICOUT 65
 7AL MODEL WITH AN EQUILIBRIUM BOUNDARY/20X.41H LAYER (BTU/SQ FT-SEC)OUT 66
 8FT*0.5 FAY-RIDDELL/127H QEHE = STAGNATION-POINT HEAT FLUX TIMOUT 67
 9ES SORT(R) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICAL MODEL WITH A FOUT 68
 \$ROZEN BOUNDARY LAYER/20X.41H(BTU/SQ FT-SEC)FT*0.5 FAY-RIDDDOUT 69
 \$ELL) OUT 70
 70 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 71
 71 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 72
 72 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 73
 73 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 74
 74 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 75
 75 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 76
 76 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 77
 77 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 78
 78 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 79
 79 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 80
 80 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 81
 81 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 82
 82 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 83
 83 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 84
 84 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 85
 85 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 86
 86 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 87
 87 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 88
 88 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 89
 89 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 90
 90 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 91
 91 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 92
 92 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 93
 93 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 94
 94 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 95
 95 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 96
 96 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 97
 97 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 98
 98 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 99
 99 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 100
 100 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 101
 101 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 102
 102 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 103
 103 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 104
 104 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 105
 105 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 106
 106 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 107
 107 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 108
 108 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 109
 109 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 110
 110 FROZEN SHOCK (ATM)/50H PW = SURFACE PRESSURE ON A WEDGE MODEOUT 111

ORST TRANSVERSE DIMENSION OF CHANNEL (INCHES)/93H X = AXIAL DOUT 111
 71STANCE ALONG NOZZLE, MEASURED FROM THROAT AND POSITIVE DOWNSTREAMOUT 112
 8 (INCHES)/67H XSN = N AVERAGED OVER SEVERAL PRECEDING POINTS OOUT 113
 9F THE SOLUTION/85H XW = DISTANCE FROM LEADING EDGE OF WEDGE, OUT 114
 \$MEASURED ALONG WEDGE SURFACE (INCHES)/115H YS = ORDINATE OF SOUT 115
 \$HOCK FROM LINE PARALLEL TO FREE STREAM FLOW, PASSING THROUGH LEADIOUT 116
 \$COORDINATE OF WEDGE (INCHES)/71H ZETA = NONDIMENSIONAL STREAMWISE OUT 117
 \$COORDINATE IN BLUNT WEDGE ANALYSIS) OUT 118
 END OUT 119-

ORIGINAL PAGE IS
 OF POOR QUALITY

```

SUBROUTINE OUT1
  OUTPUT ROUTINE
  LOGICAL ERR,DATAP, SUPGO,NOTRAN
  REAL ACOM(30),ELMENT(10),HP(20)
  DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
  COMMON AA(22,24),CDIJ(20,10),CAPX(20)
  COMMON A, AFNTS, AFNX, AMACH,
1 C, CARB, CH, AR, ARBA, ARBB, BZERO,
2 CRRB, CRS, CSTA, CT, CLNT, CMA, CRA, CRP,
3 CTP, CTPL, CTT, CX, CTAP, CTC, CTMAX, CTMXX,
4 DELI2,DELTAX, DLOGA, DLOGR, PRESTH, DT, ENT, FLUX, HDELX,
5 PCTEST, PRES, RHOC, RHOP, RHTH, ROBARA, ROBARP, RHO,
6 RHOBAR, RHOC, RHOP, RHTH, ROBARA, ROBARP, RHO,
7 SEN, SHPG, SC, SL, SL64, SM, SOT,
8 TEST, TESTB, TPRINT, TSTOP, UP, ZPA, SUMG,
  COMMON BE(64), BETA(64,20), BLBK(31), CAI(64), CAPXTH(20),
1 CCPJ(20), CEACT(64), CGI(20), CGMU(20), CHII(20),
2 CLNIMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJ(20), GJA(20),
3 GJB(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), OM(20),
4 QQ(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20),
5 SKIL(20), SS(20), TB(30), TFA(20), TFB(20), TFC(20),
6 TFD(20), TFE(20), TFK(20), THEV(20), XMJAT(20), XNUI(64)
  COMMON BETA(64,20), ELJ(10,20), GELJ(10,20),
1 XNUIJ(64,20),XNUIJP(64,20)
  COMMON IC, INEQ, INEQV, IP, IRUN, ISC, ISCP1,
1 ISMC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
3 ISW5B, ISW6A, ISW6B, IUPD, IZWR, JJK, LC, M1, NFIT,
4 NIT, NNN, NNS, NQS, NQT, NTEST
  COMMON IGJ(20), IGM(20), ITB(5), KUR(64,20), LPIJ(20,10)
  COMMON ACOM,ELMENT,HP
  DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBOU
1 J(20),SDGJ(20),SHJAP(20),THEVP(20)
  EQUIVALENCE (AAA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),
1 (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SEJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
  COMMON /RDOU/ FLOW,FACNAM,CHANAM,LIMOUT
  COMMON /TEMPRY/ SAVEC(20)
  COMMON /BLOUT/ REPF,THETA(2),SN(2),XSN(2),PRREF,HR,QWDOT(2),
1 TAUW(2)
  COMMON /CONVRT/ CF(6)
  DIMENSION VARNAM (35),DLGDX(20)
  EQUIVALENCE (DLGDX(1),AA(1,1))
  DATA VARNAM /X,T,H,P,R,
1 DIAM,V,M,S,GAMMA,
2 ARATEF,REF,MW,MU,SIGMA,
3 ARAT,DELSTR,THETA,QW,TAUW,
4 HR,PRREF,STANTN,RETH,RETHTR,
5 HEIGHT,CELSR,THETA,QW,TAUW,
6 HR,PRREF,STANTN,RETH,RETHTR/
  DIMENSION VARN2(2),FV02(2,2)
  DATA VARN2 /N,XSN/
  DATA DNAM /DIAM/,WNAM /WIDTH/,ANAM /ARAT/
  COMMON /OUTPUT/ FVOUT(35),GJMF(20)

```



```

50  FAREA=2.*RTHCM(1)
60  GO TO 100
70  IF (NPRFLS.EQ.1) GO TO 80
80  DO 70 I=1,2
90  DELSTP(I)=(1.-OMDST(I))*R0
100 FAREA=4.*(RTHCM(1)-DELSTP(1))*(RTHCM(2)-DELSTP(2))
110 GO TO 100
120 IF (JDIM.EQ.0) GO TO 90
130 FAREA=PZ*(OMDST(1)*RTHCM(1))*2
140 GO TO 100
150 FAREA=2.*OMDST(1)*RTHCM(1)
160 FAREA=FAREA/929.03
170 WRITE (6,570)
180 TFLOW=FAREA*CF(1)*CF(2)*SM
190 IF (ISW2B.GT.0) GO TO 110
200 WRITE (6,580) FLOW
210 IF (IC.FQ.0) GO TO 120
220 DEL=SAVEC(1)*RHO*TB7/CM
230 GO TO 130
240 DEL=0.
250 IF (NPRFLS.EQ.2) GO TO 140
260 WRITE (6,590) FACNAM,DIAM(1)
270 GO TO 150
280 WRITE (6,600) CHANAM,DIAM(1),DIAM(2),FACNAM
290 WRITE (6,610) POUT,TOUT,HOUT,SOUT,RCUT,UCUT,FLX,TFLOW,GAMMA,CM,DEL
300 *WRITE (6,620)
310 WRITE (6,630) (HP(I),SAVEC(I),I=1,ISS)
320 RETURN
330 FVOUT(1)=CF(4)*CX
340 FVOUT(2)=CT*CTAP
350 FVOUT(3)=CF(5)*CH
360 FVOUT(4)=PRES*PRESA
370 FVOUT(5)=CF(1)*RHO
380 FVOUT(7)=CF(2)*SU
390 FVOUT(8)=AMACH
400 FVOUT(9)=SEN
410 FVOUT(11)=AFNX
420 I6=16+1
430 IF (ISW3B.EQ.0) GO TO 180
440 DO 170 I=1,NPRFLS
450 I1=7+10*I
460 FVOUT(I1)=CF(6)*DELBL(I)
470 FVOUT(I1+1)=CF(6)*THETA(I)
480 FVOUT(I1+2)=QWDOT(I)
490 FVOUT(I1+3)=TAUW(I)
500 FVOUT(I1+4)=1.8*HR
510 FVOUT(I1+5)=PRREF
520 CONTINUE
530 VARNAM(6)=DNAM
540 VARNAM(15)=ANAM
550 IF (NPRFLS.EQ.1) GO TO 200
560 CALL GVAR2 (CX,Y,Z)
570 IF (ERR) RETURN
580 IF (ISW3B.NE.0) GO TO 190
590 CHDIMS(2)=2.*Y*CF(4)

```

PRECEDING PAGE BLANK NOT FILMED

190	CHDIMS(1)=2.*Z*CF(4)	OUI 166
	FVOUT(6)=1.E20	OUI 167
	GO TO 200	OUI 168
	FVOUT(16)=2.*Z*CF(4)	OUI 169
	FVOUT(26)=2.*Y*CF(4)	OUI 170
200	VARNAM(16)=WNAME	OUI 171
	CONTINUE	OUI 172
	FVOUT(10)=GAMMA	OUI 173
	IF (ISW3B.EQ.0) GO TO 250	OUI 174
	IF (NPRFLS.EQ.2) GO TO 210	OUI 175
	CALL GMAR (CX,Y)	OUI 176
	IF (ERR) RETURN	OUI 177
	FVOUT(6)=2.*Y*CF(4)	OUI 178
210	IF (NOTRAN) GO TO 220	OUI 179
	CALL TRANSP (FVOUT(2),FVOUT(4))	OUI 180
	IF (ERR) RETURN	OUI 181
220	FVOUT(14)=VISC*CF(3)	OUI 182
	IF (VISC.EQ.0) GO TO 230	OUI 183
	FVOUT(12)=FVOUT(5)*FVOUT(7)/FVOUT(14)	OUI 184
	GO TO 240	OUI 185
230	FVOUT(12)=0.	OUI 186
240	FVOUT(15)=SIGMA	OUI 187
	GO TO 260	OUI 188
250	FVOUT(12)=REPF	OUI 189
	FVOUT(14)=CF(3)*VISC.	OUI 190
	FVOUT(15)=SIGMA	OUI 191
260	FVOUT(13)=CM	OUI 192
	H2=1.8*CHA*CRP/CMA	OUI 193
	IF (ISW3B.EQ.0) GO TO 340	OUI 194
	IF (FVOUT(7).EQ.0) GO TO 270	OUI 195
270	STFAC=1./(FVOUT(5)*FVOUT(7)*(H0-HW))	OUI 196
	FVOUT(23)=STFAC*FVOUT(19)	OUI 197
	FVOUT(24)=FVOUT(12)*FVOUT(18)/12.	OUI 198
	FVOUT(25)=200.*EXP(0.224*FVOUT(8))	OUI 199
	IF (NPRFLS.EQ.1) GO TO 280	OUI 200
	FVOUT(33)=STFAC*FVOUT(29)	OUI 201
	FVOUT(34)=FVOUT(12)*FVOUT(28)/12.	OUI 202
	FVOUT(35)=FVOUT(25)	OUI 203
280	IF (LIMOUT.EQ.0) GO TO 300	OUI 204
	DO 290 I=1,NPRFLS	OUI 205
	FVOUT(1,I)=SN(I)	OUI 206
290	FVOUT(2,I)=XSN(I)	OUI 207
300	CALL GEOMAR (CX,ARAT,DARAT)	OUI 208
	IF (ERR) RETURN	OUI 209
	IF (NPRFLS.EQ.2) GO TO 310	OUI 210
	FVOUT(16)=ARAT	OUI 211
310	GO TO 320	OUI 212
	FVOUT(6)=ARAT	OUI 213
	VARNAM(6)=ANAM	OUI 214
320	GO TO 340	OUI 215
	IF (JDIM.EQ.0) GO TO 330	OUI 216
	FVOUT(6)=RTHCM(1)*SQRT(ARAT)*CF(4)*2.	OUI 217
	GO TO 340	OUI 218
330	FVOUT(6)=RTHCM(1)*ARAT*CF(4)*2.	OUI 219
340	INEQPI=INEQ+1	OUI 220

```

350 WRITE (6,640) (ASTRSK,I=1,8),(ASOLN(J,1,CLN),J=1,3),(ASTRSK,K=1,3) OUI 221
360 1,NSTEPS,STEP(ISCLN),(ASTRSK,L=1,4),TYPOLN(INEQ1,ISOLN) OUI 222
    IF (ISW3B.EQ.0) GO TO 360 OUI 223
    DO 350 L=1,NPRFLS OUI 224
    I1=6+10*L OUI 225
    I2=I1+9 OUI 226
    WRITE (6,660) (VARNAM(I),FVOUT(I),I=1,12) OUI 227
    CONTINUE OUI 228
    IF (NPRFLS.EQ.1,CR.ISW3B.NE.0) GO TO 370 OUI 229
    WRITE (6,670) CHDIMS(1),CHDIMS(2) OUI 230
    IF (LIMOUT.EQ.0) GO TO 380 OUI 231
    WRITE (6,680) ((VARN2(I),J,FVOUT(I),I=1,2),J=1,NPRFLS) OUI 232
    IF (NT.EQ.1,CR.ISOLN.NE.3) GO TO 390 OUI 233
    TNOUT(1)=CTAP*CTE OUI 234
    TNOUT(2)=ODPR OUI 235
    TNOUT(3)=ODPE OUI 236
    TNOUT(4)=H0 OUI 237
    WRITE (6,690) (TNAM(I),TNOUT(I),I=1,4) OUI 238
    IF (ISW68.EQ.0,CR.ISOLN.EQ.1) GO TO 440 OUI 239
    IF (I6.LT.IS6) GO TO 440 OUI 240
    I6=0 OUI 241
    WRITE (6,690) GO TO 400 OUI 242
    IF (IC.EQ.0) GO TO 400 OUI 243
    EMF=SAVEC(1) OUI 244
    SAVEC(1)=EMF*RHO*TB7/CM OUI 245
    ISSSP1=(ISS-1)/5+1 OUI 246
    DO 410 K=1,ISSSP1 OUI 247
    LIM1=5*(K-1)+1 OUI 248
    LIM2=MINO(5*K,ISS) OUI 249
    WRITE (6,700) (H0(I),SAVEC(I),I=LIM1,LIM2) OUI 250
    IF (IC.EQ.0) GO TO 420 OUI 251
    SAVEC(1)=EMF OUI 252
    IF (ISW58.GT.0,CR.ISOLN.NE.3) GO TO 440 OUI 253
    WRITE (6,510) OUI 254
    WRITE (6,520) (I,PI(I),I=1,ISR) OUI 255
    WRITE (6,530) (I,CHI(I),I=1,ISR) OUI 256
    WRITE (6,540) (I,PCHI(I),I=1,ISR) OUI 257
    DO 430 I=1,ISS OUI 258
    DLGDX(I)=DGJ(I)/GJ(I) OUI 259
    WRITE (6,550) (HP(I),DLGDX(I),I=1,ISS) OUI 260
    IF (.NOT.DATAP) GO TO 500 OUI 261
    IF (ITYPER.NE.1) GO TO 450 OUI 262
    IMP=0 OUI 263
    IX=0 OUI 264
    IX=IX+1 OUI 265
    NRCOUT=NRCOUT+1 OUI 266
    ITYPER=2 OUI 267
    XXX(1)=FVOUT(1) OUI 268
    XXX(2)=FVOUT(2) OUI 269
    XXX(3)=FVOUT(3) OUI 270
    XXX(4)=FVOUT(4) OUI 271
    XXX(5)=FVOUT(5) OUI 272
    XXX(6)=FVOUT(6) OUI 273
    XXX(7)=FVOUT(7) OUI 274
    XXX(8)=FVOUT(8) OUI 275

```

```

460      IF (ISW3B) 460,480,460
      IBL=17+10*(NBL-1)
      XXX(9)=FVOUT(1BL)
      IF (NPRELS.EQ.2) GO TO 470
      XXX(8)=FVOUT(16)
      GO TO 490
      XXX(8)=FVOUT(6)
      GO TO 490
      XXX(8)=FVOUT(11)
      XXX(9)=0.
      XXX(10)=FVOUT(12)
      WRITE (1TOUT) ITPER,IX,(XXX(I),I=1,10)
      RETURN
C
C
C
510      FORMAT (57X,18HREACTION RATE DATA)
520      FORMAT (3H P.,11X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)
530      FORMAT (4H CHI,10X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)
540      FORMAT (6H PICH,11X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)
550      FORMAT (4H DLG,12X,7(2X,A4,1PE10.1)/8(2X,A4,1PE10.1),4X)
560      FORMAT (1X,A3,1H(.11,3H) =.4X,1PE10.3,4X,3(A3,1H(.11,3H) =.4X,E.0.
      13,4X))
570      FORMAT (1H0)
580      FORMAT (19H GAS FLOW RATE =.F10.3,2X,7H LB/SEC)
590      FORMAT (10H NOZZLE - .A4,5X,F10.3,2X,21H INCH THROAT DIAMETER)
600      FORMAT (9H CHANNEL .A4,6X,F10.3,3H BY,F7.3,17H INCH THROAT FOR .A4
      1)
610      FORMAT (19H PRESSURE =.F1.3,2X,4H ATM/19H TEMPERATURE
      1 =.F7.0,5X,6H DEG K/19H ENTHALPY =.F7.0,5X,7H BTU/LB,.9H
      2ENTROPY =.F9.2,3X,13H BTU/LB-DEG R/19H DENSITY =.F7.0,5X,7H FT/SEC/19H
      3,FI2,5,9H LB/CU FT/19H VELOCITY =.F7.0,5X,7H FT/SEC/19H MAQUI
      4SS FLUX =.F10.3,2X,13H LB/SQ FT-SEC/19H COMPUTED FLOW =.F10.3,2X,
      5F10.3,2X,7H LB/SEC/19H GAMMA =.F10.3,19H MOLECULAR WEIGHT/19H
      6HT =.F9.2,3X,8H GM/MOLE/19H ELECTRON DENSITY =.3X,1PE9.2,13H ELECTRO
      7RONS/CC)
620      FORMAT (/10X,22HSPECIES MOLE FRACTIONS/)
630      FORMAT (13X,A4,4X,1PE10.3)
640      FORMAT (/6H ***,14A6,16,6A6//)
650      FORMAT (1X,A6,2H =.F11.3,7X,A6,2H =.F8.0,10X,A6,2H =.F8.0,10X,A6,2H
      1H =.4X,1PE10.3,4X,A6,2H =.4X,E10.3/1X,A6,2H =.0PF11.3,7X,A6,2H =.F0U1
      28.0,10X,A6,2H =.F11.3,7X,A6,2H =.F10.2,8X,A6,2H =.F11.3/1X,A6,2H =.F0U1
      3,FI1.3,7X,A6,2H =.4X,1PE10.3,4X,A6,2H =.0PF10.2,8X,A6,2H =.4X,1PE10.3
      4,3,4X,A6,2H =.4X,F10.3)
660      FORMAT (1X,A6,2H =.F11.3,7X,A6,2H =.F11.3,7X,A6,2H =.F11.3,7X,A6,2H
      1H =.4X,1PE10.3,4X,A6,2H =.4X,E10.3/1X,A6,2H =.0PF8.0,10X,A6,2H =.F0U1
      212.4,6X,A6,2H =.4X,1PE10.3,4X,A6,2H =.0PF8.0,10X,A6,2H =.F8.0)
670      FORMAT (9X,WIDTH =.F11.3,6X,9H HEIGHT =.F11.3)
680      FORMAT (1X,A6,2H =.F8.0,10X,A6,2H =.4X,1PE10.3,4X,A6,2H =.4X,E10.3
      1,4X,A6,2H =.4X,F10.3)
690      FORMAT (45X,41HSPECIES MOLE FRACTIONS IN THE FREE STREAM)
700      FORMAT (5(1X,.4,3X,1H=.4X,1PE10.3,3X))
      END

```

```

SUBROUTINE PERT
LOGICAL ERR
REAL ACOM(30),ELMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON C, CARB, CSTA, CTT, DLOGR, PRESB, RHOP, RHOC, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
1 CRRB, CRS, CTA, CTT, DLOGR, PRESB, RHOP, RHOC, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
2 CTP, CTPL, DLOGA, PRESA, RHOP, RHOC, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
3 DELT2, DELTAX, DLOGA, PRESA, RHOP, RHOC, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
4 SPTTEST, PRES, RHOC, RHOP, RHOC, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
5 RMDBAR, RHOC, RHOP, RHOC, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
6 SEN, SHPG, SC, TSTOP, TPRINT, TFE(20), TFC(20), XNUI(64),
7 TEST, TSTB, TPRINT, TFE(20), TFC(20), XNUI(64),
8 COMMON BE(64), CACT(64), CLNPI(64), PERGJ(20), PGJ(20), SDCHI(64), TB(30), TFA(20), THEV(20), ELJ(10,20),
COMMON CCPJ(20), CACT(64), CLNPI(64), PERGJ(20), PGJ(20), SDCHI(64), TB(30), TFA(20), THEV(20), ELJ(10,20),
1 CCJ(20), CACT(64), CLNPI(64), PERGJ(20), PGJ(20), SDCHI(64), TB(30), TFA(20), THEV(20), ELJ(10,20),
2 CLNIMC(64), CLNPI(64), PERGJ(20), PGJ(20), SDCHI(64), TB(30), TFA(20), THEV(20), ELJ(10,20),
3 GJ(20), PERGJ(20), PGJ(20), SDCHI(64), TB(30), TFA(20), THEV(20), ELJ(10,20),
4 GJ(64), SAJ(20), SDCHI(64), TB(30), TFA(20), THEV(20), ELJ(10,20),
5 SKIL(20), SS(20), TFE(20), TFC(20), XNUI(64),
6 TFD(20), TFE(20), TFC(20), XNUI(64),
COMMON XNUIJ(64,20),XNUIJP(64,20),
1 COMMON IC, IM, INEQV, IP, IRUN, ISC, ISCP1,
1 ISMC, ISVCR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
3 ISW5B, ISW6A, ISW6B, IUPD, IZERQ, JJK, LC, M1, NFIT,
4 NIT, NNN, NNS, NQS, NQT, NTEST, KUR(64,20), LRIJ(20,10),
COMMON IGM(20),
COMMON ACOM,ELMENT,HP
DIMENS:ON AAA(22,24),RTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SB
1J(20),SDGJ(20),SHJAP(20),THEVP(20)
EQUIVALENCE (AAA(1,1),AAA(1,1)),(RTA(1,1),RTA(1,1)),(GJA(1,1),GJA(1,1)),
1 (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1,1),DGJ(1,1)),
2 (APX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1,1)),
3 (JA(1),SHJAP(1)),(THEV(1),THEVP(1))
COMMON /ERROR/ ERR
DO 10 I=1,ISSP2
DO 10 J=1,ISSP3
AA(I,J)=0.
DO 20 I=1,ISC
DO 20 J=1,ISS
AA(I,J)=LPIJ(J,I)
DO 40 I=ISCP1,ISS
DO 40 J=1,ISS
Z=0.
DO 30 K=1,ISR
Z=Z+(BETA(K,I)*PI(K)*BETA(K,J))
Z=Z/GJ(J)
AA(I,J)=-Z
DO 50 J=1,ISS
AA(I,ISSP1,J)=(CMA*SHJ(J)/(SU2))
DO 80 I=ISCP1,ISS
ZI=0.
DO 70 K=1,ISR

```

60	Z=0.		56	PER
70	DO 60 J=1,ISS		57	PER
80	Z=Z+(BETA(K,J))*((SHJ(J)/CT)-1.)		58	PER
	Z1=Z1+Z*(BETA(K,I)*PI(K))		59	PER
	AA(I,ISSP1)=Z1/CT		60	PER
	Z=0.		61	PER
90	DO 90 J=1,ISS		62	PER
	Z=Z+GJ(J)*CCPJ(J)		63	PER
	AA(ISSP1,ISSP1)=(CMA*Z)/(SU2)		64	PER
	DO 110 I=ISSP1,ISS		65	PER
	Z1=0.		66	PER
	DO 100 K=1,ISR		67	PER
100	Z1=Z1+(BETA(K,I)*PI(K)*BE(K))		68	PER
110	AA(I,ISSP2)=-Z1/RHO		69	PER
	AA(ISSP1,ISSP2)=-1./RHO		70	PER
	DO 120 I=1,ISS		71	PER
120	AA(ISSP2,I)=SENT(I)-1.0-ALOG(GJ(I)*RHO*CT*82.055*RHAP*CTAP)		72	PER
	Z=0.		73	PER
	DO 130 J=1,ISS		74	PER
130	Z=Z+GJ(J)*((CCPJ(J)-1.)		75	PER
	AA(ISSP2,ISSP1)=Z/CT		76	PER
	AA(ISSP2,ISSP2)=-1./(CM*RHO)		77	PER
	DO 140 I=ISSP1,ISS		78	PER
140	AA(I,ISSP3)=DGJ(I)		79	PER
	CALL DSMSQL(AA,ISSP2,22)		80	PER
	IF (ERR) RETURN		81	PER
	DO 150 I=1,ISS		82	PER
150	PERTGJ(I)=AA(I,ISSP3)		83	PER
	PCT=AA(ISSP1,ISSP3)		84	PER
	PRHO=AA(ISSP2,ISSP3)		85	PER
	DO 170 I=1,ISR		86	PER
	Z=0.		87	PER
	Z1=0.		88	PER
	DO 160 J=1,ISS		89	PER
	Z=Z+BETA(I,J)*((SHJ(J)/CT)-1.)		90	PER
160	Z1=Z1+(BETA(I,J)*PERTGJ(J))/GJ(J)		91	PER
170	SOCHI(I)=-BE(I)*PRHO/RHO+Z*PCT/CT-Z1		92	PER
	RETURN		93	PER
	END		94-	PER

```

10  FUNCTION PIOMEG(TE)
      COMMON /TNEG/ TLIST(30), POM(30)
      DO 10 I=2,30
      IF (TE.LT.TLIST(I)) GO TO 20
      CONTINUE
      PIOMEG=POM(30)
      GO TO 30
20  PIOMEG=POM(I-1)+(POM(I)-POM(I-1))*(TE-TLIST(I-1))/(TLIST(I)-TLIST(I-1))
30  RETURN
      END

```

1
2
3
4
5
6
7
8
9
10
11-

PI0
PI0
PI0
PI0
PI0
PI0
PI0
PI0
PI0
PI0
PI0

```

SUBROUTINE PROP
REAL ACOM(30),ELMENT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A,AFNTS,AFNX,AMACH,
C,CARB,CST,CH,
1 CRRB,CRS,CSTA,CTT,CX,
2 CTP,CTPL,DLOGA,DLOGR,PRESA,PRHOC,
3 DELT2,DELTAX,DLOGA,DLOGR,PRESA,PRHOC,
4 SPCTEST,PRHOC,PRHOC,PRHOC,PRHOC,PRHOC,
5 RHOBAR,PRHOC,PRHOC,PRHOC,PRHOC,PRHOC,
6 SEN,SHPG,SC,SL,SL64,
7 TEST,TESTB,TPRINT,TSTOP,UP,
8 COMMON BE(64),BET(20),BLBK(31),
1 CCRPJ(20),CEACT(64),CGI(20),CGMU(20),
2 CLNMC(64),CLNPI(64),CMW(20),ETAI(64),
3 GJR(20),PERTGJ(20),PGJ(20),PI(64),
4 QO(64),SAJ(20),SDCHI(64),SENT(20),
5 SKIL(20),SS(20),TFE(20),TFK(20),
6 TFD(20),TFC(20),THEV(20),THEVP(1),
COMMON BETA(64,20),ELJ(10,20),GELJ(10,20),
1 XNUIJ(64,20),XNUIJP(64,20),
COMMON IC,INEQ,INEQV,IP,IRUN,ISC,ISCP1,
1 ISMC,ISMCNR,ISR,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,
2 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,ISW5A,
3 ISW5B,ISW6A,ISW6B,IUPD,IZERO,JJK,LC,M1,NFIT,
4 NIT,NNN,NNS,NQS,NQI,NTEST
COMMON ACOM,ELMENT,HP
DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),GJ(20),SBP
1 J(20),SDGJ(20),SHJAP(20),THEVP(20)
EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),
1 (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
DATA RNAME /4HPROP/
S4=0.
SHPG=0.
DO 10 J=1,ISS
S4=S4+GJA(J)*SENT(J)
SHPG=SHPG+GJA(J)*SHJ(J)
ZP=CMA*(S4+ENT)
IF (ZP.LT.80.) GO TO 20
CALL DUMP (RNAME)
RETURN
PRES=EXP(ZP)
CH=CMA*SHPG
RHOBAR=PRES/CT
RHO=(RHOBAR*RHOBAR)/(1.+RHOBAR*ROBARP*BZERO)
CH=CH+(PRES/RHOBAR)*(RHOBAR*ROBARA/RHO-1.0)
SU2=2.*(CHA-CH)
SU=SORT(SU2)
FLUX=RHO*SU
RETURN
END

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

IF (ERR) RETURN
NSTEPS=0
IF (.NOT. MODLPT) GO TO 30
CALL MODEL
MODLPT=.FALSE.
IF (INEQ.NE.0) GO TO 40
CT=CT-PCT
CTE=CT
RHO=RHO-PRHO
RETURN
END

```

30

40
C

```

PRT 56
PRT 57
PRT 58
PRT 59
PRT 60
PRT 61
PRT 62
PRT 63
PRT 64
PRT 65
PRT 66
PRT 67-

```

```

SUBROUTINE PUTQIN (X)
SUBROUTINE PUTQIN SUPERVISES THE COMPUTATION OF COLLISION CROSS
SECTIONS FOR THE TRANSPORT CALCULATIONS.
LOGICAL WROTE
DIMENSION OM(3)
COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ
COMMON /TRANS8/ NV(15),N
COMMON /RDTR/ ISW8B
DATA IDBG /-1/
IDBG=IDBG+1
WROTE=.FALSE.
INITIALIZE Q ARRAY TO ZERO.
DO 10 I=1,N
DO 10 J=1,N
DO 10 K=1,3
O(K,I,J)=0.
CALL QCOUL (OM,X,V)
O(1,2,1)=OM(1)
COMPUTE CROSS SECTION VALUES AND ACCUMULATE IN Q ARRAY.
M=2
LQ1=1
DO 240 L=1,NKQ
K=KQ(L)
LQ2=NQ(L)
GO TO (20,30,40,50,60,70,80,90,100,110,120,130,140,150,160), K
CONTINUE
GO TO 190
OM(1)=V(M)*Q(1,2,1)
OM(3)=OM(1)*1.5625
OM(2)=V(M+1)*OM(1)
GO TO 190
CALL QEXP (OM,V(M))
GO TO 190
CALL QEX (OM,V(M))
IF (V(M+3)) 190,190,170
CALL QIAB (OM,V(M))
GO TO 190
CALL GREPP (OM,V(M))
GO TO 190
CONTINUE
17 CALL QATTP (OM,V(M))
GO TO 190
CALL QLJ (OM,V(M))
GO TO 190
CALL QSAME (OM,V(M))
GO TO 190
CALL QMIX (LQ1,LQ2)
GO TO 210
CALL Q11 (LQ1,LQ2,V(M))
GO TO 210
CALL Q12 (OM,V(M))
GO TO 190
CALL Q13 (LQ1,LQ2,V(M))
GO TO 210

```

```

15 CALL Q14 (LQ1,LQ2,V(M))
GO TO 210
160 CONTINUE
GO TO 210
170 DO 180 LQ=LQ1,LQ2
I=IQ(LQ)
J=JQ(LQ)
Q(I,I,J)=OM(1)
Q(3,I,J)=OM(3)
180 GO TO 210
190 DO 200 LQ=LQ1,LQ2
I=IQ(LQ)
J=JQ(LQ)
DO 200 KK=1,3
Q(KK,I,J)=Q(KK,I,J)+OM(KK)
200 M=M+NV(K)
210 C
DEBUG OUTPUT
IF (ISW8.EQ.0) GO TO 240
IF (IDBG.EQ.0) GO TO 220
IF (IDBG.NE.ISW8) GO TO 240
220 WRITE=TRUE
WRITE (6,250) L,K,NQ(L),M,LQ1
DO 230 I=1,N
WRITE (6,260) (Q(I,J,I),J=1,I)
230 LQ1=LQ2+1
240 IF (WRN) IDBG=0
RETURN
C
250 FORMAT (1HC/1015/)
260 FORMAT (1P10E12.4)
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

PUT 56
PUT 57
PUT 58
PUT 59
PUT 60
PUT 61
PUT 62
PUT 63
PUT 64
PUT 65
PUT 66
PUT 67
PUT 68
PUT 69
PUT 70
PUT 71
PUT 72
PUT 73
PUT 74
PUT 75
PUT 76
PUT 77
PUT 78
PUT 79
PUT 80
PUT 81
PUT 82
PUT 83
PUT 84
PUT 85
PUT 86-

```

SUBROUTINE QCOUL (OM1,X,C)
 SUBROUTINE QCOUL COMPUTES THE FACTOR 0.8*QC IN THE EFFECTIVE
 COULOMB CROSS SECTION.
 COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
 OM1=2.2346E10/T**2
 IF (C) 20,20,10
 Y=(T**4)/(1.E15*X+1.E-35*(T**4))
 OM1=.5*OM1*ALOG(2.091E1*Y+1.517E2*Y**.666667)
 RETURN
 END

C

10

20

QCO
 QCO
 QCO
 QCO
 QCO
 QCO
 QCO
 QCO
 QCO
 QCO
 10-

1

2

3

4

5

6

7

8

9

10-

SUBROUTINE QEX (OM,V)
 SUBROUTINE QEX COMPUTES CROSS SECTIONS FOR A RESONANT EXCHANGE
 PROCESS.
 DIMENSION OM(3),V(3)
 COMMON /TRANSI/ T,Q(3,20,20),ZM2(20)
 A=V(1)
 B=V(2)
 ZM=V(3)
 Y=A-.217147*B*ALOG(T/ZM)
 OM(1)=2.*(Y-4.461*B)**2+.03724*B**2
 OM(3)=2.*(Y-4.317*B)**2-.06754*B**2
 OM(2)=0.
 RETURN
 END

1 2 3 4 5 6 7 8 9 10 11 12 13 14-
 QEX QEX QEX QEX QEX QEX QEX QEX QEX QEX QEX QEX QEX QEX

ORIGINAL PAGE
 OF POOR QUALITY

SUBROUTINE QEXP (OM,V)
 SUBROUTINE QEXP COMPUTES CROSS SECTIONS FOR AN EXPONENTIAL
 INTERACTION POTENTIAL.

DIMENSION OM(3),V(3)
 COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
 A=V(1)
 RHD=V(2)
 N1=V(3)
 ALPHA=ALOG(A/T)
 CALL QINTRP (ALPHA,OM,N1,50)
 OM(1)=12.5664*(ALPHA*RHD)**2*OM(1)
 OM(2)=OM(2)*OM(1)
 OM(3)=OM(3)*OM(1)
 RETURN
 END

ORIGINAL PAGE IS
 OF POOR QUALITY

QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP
 QXP

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15-

C	SUBROUTINE QINTRP (A,B,N1,N)	QIN
C	SUBROUTINE QINTRP INTERPOLATES TABULAR DATA FOR THE CROSS SECTION	QIN
	COMPUTATIONS	QIN
	COMMON /ROTR/ ISW8B	QIN
	DIMENSION B(3)	QIN
	COMMON /TRANS4/ TL(1000),OM(1000,3)	QIN
	I=N1	QIN
10	IF (TL(I)-A) 10,40,30	QIN
	N2=N1+N-2	QIN
	DO 20 I=N1,N2	QIN
	IF (TL(I+1))-A; 20,40,40	QIN
20	CONTINUE	QIN
	I=N2	QIN
30	IF (ISW8B.EQ.0) GO TO 40	QIN
	WRITE (6,60) I,A,TL(I),TL(I+1)	QIN
40	DO 50 J=1,3	QIN
50	B(J)=OM(I+1,J)+(A-TL(I+1))*OM(I,J)-OM(I+1,J))/(TL(I)-TL(I+1))	QIN
C	RETURN	QIN
	FORMAT (25HC EXTRAPOLATED VALUES I6,1P4E14.4/1H)	QIN
60	END	QIN

SUBROUTINE QLJ (OM,V)
 SUBROUTINE QLJ COMPUTES CROSS SECTIONS FOR THE LENNARD-JONES 6-12
 POTENTIAL.
 DIMENSION OM(3), V(3)
 COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
 EPSLN=V(1)
 SIGMA=V(2)
 N1=V(3)
 TSTAR=T/EP SLN
 CALL QINTRP (TSTAR,OM,N1,37)
 OM(1)=3.1416*SIGMA**2*OM(1)
 OM(2)=OM(2)*OM(1)
 OM(3)=OM(3)*CM(1)
 RETURN
 END

QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ
 QLJ

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15-

ORIGINAL PAGE IS
 OF POOR QUALITY

```

C SUBROUTINE QMIX (LQ1,LQ2)
C SUBROUTINE QMIX COMPUTES CROSS SECTIONS FROM THE EMPIRICAL MIXING
C RULE
C
C DIMENSION SQT(3,20)
C COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
C COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKQ
C IF (LQ2.LT.LQ1) GO TO 70
C DO 10 I=1,20
C   SQR(I,I)=0.
C   DO 60 LQ=LQ1,LQ2
C     I=IQ(LQ)
C     J=JQ(LQ)
C     IF (SQT(1,I).GT.0.) GO TO 30
C     DO 20 K=1,3
C       SQT(K,I)=5*SQRT(Q(K,I,I))
C     IF (SQT(1,J).GT.0.) GO TO 50
C     DO 40 K=1,3
C       SQT(K,J)=5*SQRT(Q(K,J,J))
C     DO 60 K=1,3
C       Q(K,I,J)=(SQT(K,I)+SQT(K,J))*2+Q(K,I,J)
C     RETURN
C   END
C 10
C 20
C 30
C 40
C 50
C 60
C 70

```

```

SUBROUTINE GREPP (OM,VV)
SUBROUTINE GREPP COMPUTES CROSS SECTIONS FOR AN INVERSE POWER LAW
INTERACTION POTENTIAL.
DIMENSION OM(3),VV(3)
COMMON /TRANS1/ T,O(3,20,20),ZM2(20)
COMMON /TRANS4/ TL(1000),OMEGA1(1000),ASTAR(1000),BSTAR(1000)
ITL=VV(1)
OM(1)=OMEGA1(ITL)*T**(-2./VV(2))
OM(2)=ASTAR(ITL)*OM(1)
OM(3)=BSTAR(ITL)*OM(1)
RETURN
END

```

GRE
 GRE
 GRE
 GRE
 GRE
 GRE
 GRE
 GRE
 GRE
 GRE
 GRE

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12-

```

SUBROUTINE QSAME (QM,VV)
SUBROUTINE QSAME SETS THE CROSS SECTION FOR A GIVEN SPECIES PAIR
EQUALLY TO THOSE FOR ANOTHER PAIR.
DIMENSION QM(3),VV(5)
JMMCN / TRANS1/ T,Q(3,20,20),ZM2(20)
I=VV(1)
J=VV(2)
C=VV(3)
QM(1)=C*Q(1,I,J)
QM(2)=C*Q(2,I,J)*VV(4)
QM(3)=C*Q(3,I,J)*VV(5)
RETURN
END

```

1 QSA
 2 QSA
 3 QSA
 4 QSA
 5 QSA
 6 QSA
 7 QSA
 8 QSA
 9 QSA
 10 QSA
 11 QSA
 12 QSA
 13- QSA

SUBROUTINE QTAB (OM,V)
 USE TABULAR DATA FOR CROSS SECTIONS.
 DIMENSION CM(3),V(3)
 COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
 A=V(1)
 N1=V(2)
 NLEV(3)
 CALL QINTRP (T,CM,N1,NL)
 CM(1)=A*CM(1)
 CM(2)=A*CM(2)
 CM(3)=OM(3)*OM(1)
 RETURN
 END

ORIGINAL PAGE IS
 OF POOR QUALITY

1 2 3 4 5 6 7 8 9 10 11 12 13-
 QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA
 QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA QTA

```

SUBROUTINE Q11 (L01,L02,VI)
SUBROUTINE Q11 MULTIPLIES CROSS SECTIONS BY A RAMP FUNCTION OF
TEMPERATURE.
DIMENSION VV(3),VI(2)
COMMON /TRANS1/ T.O(3,20,20),ZM2(20)
DATA VV/3*1./
T0=VI(1)
T1=VI(2)
VV(1)=AMAX1(0.,(T-T0)/(T1-T0))
IF (VV(1).LT.1.) CALL Q14 (L01,L02,VV)
RETURN
END

```

011	1
011	2
011	3
011	4
011	5
011	6
011	7
011	8
011	9
011	10
011	11
011	12-

C

10

```

SUBROUTINE Q12 (QM,VV)
SUBROUTINE Q12 COMPUTES CROSS SECTIONS FROM THE GENERALIZED MIXING
RULE (EQ. 27),
DIMENSION QM(3),VV(4)
COMMON /TRANS/ T,Q(3,20,20),ZM2(20)
I=VV(1)
J=VV(2)
K=VV(3)
L=VV(4)
DO 10 M=1,3
QM(M)=.25*(SQRT(Q(M,I,J))+SQRT(Q(M,K,L)))*#2
RETURN
END

```

```

1 2 3 4 5 6 7 8 9
012 012 012 012 012 012 012 012 012
10 11 12 13-
012 012 012 012

```

C

10

SUBROUTINE Q13 (LQ1,LQ2,VV)
 SUBROUTINE Q13 SETS TWO CROSS SECTIONS EQUAL FOR A SPECIES PAIR.

DIMENSION VV(3)
 COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
 COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JQ(400),NKO
 K1=VV(1)
 K2=VV(2)
 C=VV(3)
 DO 10 LQ=LQ1,LQ2
 I=IQ(LQ)
 J=JQ(LQ)
 Q(K1,I,J)=C*Q(K2,I,J)
 RETURN
 END

10

013 1
 013 2
 013 3
 013 4
 013 5
 013 6
 013 7
 013 8
 013 9
 013 10
 013 11
 013 12
 013 13
 013 14-


```

SUBROUTINE Q14 (LQ1,LQ2,VV)
SUBROUTINE Q14 MULTIPLIES CROSS SECTIONS BY A CONSTANT.
DIMENSION VV(3)
COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JN(400),NKO
C2=VV(1)*VV(2)
C3=VV(1)*VV(3)
DO 10 LQ=LQ1,LQ2
I=IQ(LQ)
J=JO(LQ)
Q(1,I,J)=VV(1)*Q(1,I,J)
Q(2,I,J)=C2*Q(2,I,J)
Q(3,I,J)=C3*Q(3,I,J)
RETURN
END

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15-
 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14 Q14

ORIGINAL PAGE
 OF POOR QUALITY

```

SUBROUTINE RADIUS ( ITYPE,X,R,AG,AGJ,L)
DIMENSION YZ(2),NPR(2)
COMMON /BL/ DELBL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
1 SW,R0,JOIM,IPGINT
COMMON /BLRAD/ YOZO
COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCM(2),NSECT(2),
1 NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL
DATA NPR /2,1/
GO TO (10,10,40), ITYPE
CALL GMAR (X,R)
RATIO=R/R0
GO TO (20,30,30), ITYPE
AG=RATIO
AGJ=1.
GO TO 50
AG=RATIO**2
AGJ=AG
GO TO 50
CALL GMAR2 (X,YZ(1),YZ(2))
AG=YZ(1)*YZ(2)/YOZO
R=YZ(L)
M=NPR(L)
AGJ=YZ(M)**2
RETURN
END

```

10

20

30

40

50

1 RAD
2 RAD
3 RAD
4 RAD
5 RAD
6 RAD
7 RAD
8 RAD
9 RAD
10 RAD
11 RAD
12 RAD
13 RAD
14 RAD
15 RAD
16 RAD
17 RAD
18 RAD
19 RAD
20 RAD
21 RAD
22 RAD
23 RAD
24 RAD
25-

ORIGINAL PAGE 15
OF POOR QUALITY

```

SUBROUTINE READ
LOGICAL SURGO, READXS, DATAPE, READG, AAMS, SETIGS
LOGICAL WEDGEM, AXISYM, AXIMOD, NOTRAN
REAL SHAPD(4,3), SPNAME(30)
REAL ACOM(30), ELMENT(10), HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22,24), CDIJ(20,10), CAPX(20)
COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO,
C, CARB, CH, CSTA, CT, CTAP, CM, CMA, CRA, CRP,
CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT1,
DELT2, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT,
SPCTEST, PRES, PRESA, RHOC, RHOP, RHPL, RHTH, ROBARA, ROBARP, RHO, RHOB,
SRHOBAR, RHOC, RHOP, SC, SHPG, SL, SL64, SM, SU, SDT,
SEN, TESTB, TPRINT, TSTOP, UP, ZPA, SUMG,
COMMON BE(64), BCBK(31), CAI(64), CAPXTH(20),
CCPJ(20), CEACT(64), CGI(20), CGMU(20), CHI(64), CHI1(20),
2CLNIMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJ(64), GJA(20),
3GJB(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), QM(20),
4QQ(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20),
5SKIL(20), SS(20), TB(30), TFA(20), TFC(20),
6TFD(20), TFE(20), TFK(20), THEV(20), XMJAT(20),
COMMON BETA(64,20), ELJ(10,20), GELJ(10,20),
1XNUIJ(64,20), XNUIJP(64,20),
COMMON IC, INEQ, INEQV, IP, IRUN, ISC, ISCP1,
1ISMIC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISWSA,
3ISW5B, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, MI, NFIT,
4NIT, NNN, NNS, NQS, NQT, NTEST
COMMON IGJ(20), IGH(20), ITB(5), KUR(64,20), LPIJ(20,10)
COMMON ACOM, ELMENT, HP
COMMON /AREA/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1NSECTU(2), ISHAPE(12,2), NPROFL(2), NPROFLS, NBL
COMMON /AVG/ WSAVE
COMMON /BL/ DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1SN, RO, JDIM, IPCINT
COMMON /CHAN/ CPI(5), CP2(5), CP3(5), CP4(5), CP5(5)
COMMON /COLDSP/ CGMW, QPJ(10), IJCS(10), NCS
COMMON /ELEM/ EEP1(2), EEP2(2), EEP3(2), EEP4(2), EEP5(2), EEP6(2),
1EEP7(2), EEP8(2), EEP9(2), EEP10(2)
COMMON /ELFM/ EEP1(2), EEP2(2), EEP3(2), EEP4(2), EEP5(2), EEP6(2), EEP7(2),
1EP8(2), EP9(2), EP1C(2)
COMMON /LN/ ISATOM, ISMOL, JATOM, JMOL
COMMON /MASSFL/ SMASS, CTMXFI, TSTOP1, IS(20)
COMMON /MIXT/ GP1(124), GP2(124), GP3(124), GP4(124), GP5(124),
1GP6(124)
COMMON /MODPAR/ XMPI, DXMP, FSTAG, CATFAC, TMODEL, XMODP1, DXMODP,
1TPLATE, KDIM
COMMON /MOORT/ TSDIAM(20), TSAR(20), NTS, MBL
COMMON /NEWMP/ FACMP, NMODPT
COMMON /NOZZ/ ZP1(64), ZP2(64), ZP3(64), ZP4(64), ZP5(64), ZP6(64),
1ZP7(64), ZP8(64), ZP9(64), ZP10(64), ZP11(64), ZP12(64), ZP13(64),
2ZP14(64), ZP15(64), ZP16(64), ZP17(64), ZP18(64), ZP19(64),
3ZP20(64)

```

```

COMMON /POLYAT/ THEVE(4,20)
COMMON /RDLIST/ IGAS,IGASO,NOZZO,ICHANO
COMMON /RDMAIN/ HS,SUPGO,MFITER,NOTRAN
COMMON /RDMOD/ LEWIS,IAMBIP
COMMON /RDOUT/ FLOW,FACNAM,CHANAM,LIMOUT
COMMON /RDTF/ ISWB8
COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,
1    NWX,NWANGLE,RADLE,WEDGEM,AXISYM,ISWB8
COMMON /REAC/ RP1(29),RP2(29),RP3(29),RP4(29),RP5(29),RP6(29),
2    RP7(29),RP8(29),RP9(29),RP10(29),RP11(29),RP12(29),RP13(29),
3    RP14(29),RP15(29),RP16(29),RP17(29),RP18(29),RP19(29),
4    RP20(29),RP21(29),RP22(29),RP23(29),RP24(29),RP25(29),
5    RP26(29),RP27(29),RP28(29),RP29(29),RP30(29),RP31(29),
6    RP32(29),RP33(29),RP34(29),RP35(29),RP36(29),RP37(29),
7    RP38(29),RP39(29),RP40(29),RP41(29),RP42(29),RP43(29),
8    RP44(29),RP45(29),RP46(29),RP47(29),RP48(29),RP49(29),
9    RP50(29),RP51(29),RP52(29),RP53(29),RP54(29),RP55(29),
$    RP56(29),RP57(29),RP58(29),RP59(29),RP60(29),RP61(29),
$    RP62(29),RP63(29),RP64(29),RP65(29),RP66(29),RP67(29),
$    RP68(29),RP69(29),RP70(29),RP71(29),RP72(29),RP73(29),
$    RP74(29),RP75(29),RP76(29),RP77(29),RP78(29),RP79(29),
$    RP80(29),RP81(29),RP82(29),RP83(29),RP84(29),RP85(29),
$    RP86(29),RP87(29),RP88(29),RP89(29),RP90(29),RP91(29),
$    RP92(29)
COMMON /READAT/ PFESAI,DELT11,TPRNT1,DELTXI,CXMAXI,
1    CTAPI,XZERCI,PARAMI(3,12,2),BZEROI,
2    HSTAG,READXS,READG,AAMS,AXIMOD,
3    ICASE,NEELS,ISW78,INT,ICHAN,NQSI,INEQVI,NRECO,JCS(10),
4    ISCI,ISSI,ISRI,ICI,IE(10),IR(64),NSECTS(2,2),IEEP(10)
COMMON /RENE/ GAMIN,HTEST,TETEST,QTEST,DCHELL,DCHRRAT,CCHI,TTEST,
1    GTEST
COMMON /SPEC/ SPI(43),SP2(43),SP3(43),SP4(43),SP5(43),SP6(43),
2    SP7(43),SP8(43),SP9(43),SP10(43),SP11(43),SP12(43),SP13(43),
3    SP14(43),SP15(43),SP16(43),SP17(43),SP18(43),SP19(43),
4    SP20(43),SP21(43),SP22(43),SP23(43),SP24(43),SP25(43),
5    SP26(43),SP27(43),SP28(43),SP29(43),SP30(43)
COMMON /TAPOUT/ XXX(10),ITPOUT,NRCOUT,IFLOW,ITYPER,IMP,DATAP
COMMON /THRT/ RSA
COMMON /TNEC/ SUMGH,SCPGH,QDPR,QDPE
COMMON /TNE/ TN1(186),TN2(186)
COMMON /TNEQ/ TLIST(30),PCM(30)
COMMON /TNONEQ/ CFE,DTE,EPAR,EPAR(2,25),NT,ITR(25),KTF(25),
1    KTR(25),ICH(20),IPA(25)
COMMON /TRANI/
1    I1(5),I2(5),I3(5),I4(5),I5(5),
2    I6(5),I7(5),I8(5),I9(5),I10(5),
3    I11(5),I12(5),I13(5),I14(5),I15(5),
4    I16(5),I17(5),I18(5),I19(5),I20(5),
5    I21(5),I22(5),I23(5),I24(5),I25(5),
6    I26(5),I27(5),I28(5),I29(5),I30(5),
7    I31(5),I32(5),I33(5),I34(5),I35(5),
8    I36(5),I37(5),I38(5),I39(5),I40(5),
9    I41(5),I42(5),I43(5),I44(5),I45(5),
$    I46(5),I47(5),I48(5),I49(5),I50(5),
$    I51(5),I52(5),I53(5),I54(5),I55(5),
$    I56(5),I57(5),I58(5),I59(5),I60(5)

```

\$	161(5).	162(5).	163(5).	164(5).	165(5).	REA 111
\$	166(5).	167(5).	168(5).	169(5).	170(5).	REA 112
\$	171(5).	172(5).	173(5).	174(5).	175(5).	REA 113
\$	176(5).	177(5).	178(5).	179(5).	180(5).	REA 114
\$	181(5).	182(5).	183(5).	184(5).	185(5).	REA 115
\$	186(5).	187(5).	188(5).	189(5).	190(5).	REA 116
\$	191(5).	192(5).	193(5).	194(5).	195(5).	REA 117
\$	196(5).	197(5).	198(5).	199(5).	1100(5)	REA 118
\$	J1 (5).	J2 (5).	J3 (5).	J4 (5).	J5 (5).	REA 119
1	J6 (5).	J7 (5).	J8 (5).	J9 (5).	J10(5).	REA 120
2	J11(5).	J12(5).	J13(5).	J14(5).	J15(5).	REA 121
3	J16(5).	J17(5).	J18(5).	J19(5).	J20(5).	REA 122
4	J21(5).	J22(5).	J23(5).	J24(5).	J25(5).	REA 123
5	J26(5).	J27(5).	J28(5).	J29(5).	J30(5).	REA 124
6	J31(5).	J32(5).	J33(5).	J34(5).	J35(5).	REA 125
7	J36(5).	J37(5).	J38(5).	J39(5).	J40(5).	REA 126
8	J41(5).	J42(5).	J43(5).	J44(5).	J45(5).	REA 127
9	J46(5).	J47(5).	J48(5).	J49(5).	J50(5).	REA 128
\$	J51(5).	J52(5).	J53(5).	J54(5).	J55(5).	REA 129
\$	J56(5).	J57(5).	J58(5).	J59(5).	J60(5).	REA 130
\$	J61(5).	J62(5).	J63(5).	J64(5).	J65(5).	REA 131
\$	J66(5).	J67(5).	J68(5).	J69(5).	J70(5).	REA 132
\$	J71(5).	J72(5).	J73(5).	J74(5).	J75(5).	REA 133
\$	J76(5).	J77(5).	J78(5).	J79(5).	J80(5).	REA 134
\$	J81(5).	J82(5).	J83(5).	J84(5).	J85(5).	REA 135
\$	J86(5).	J87(5).	J88(5).	J89(5).	J90(5).	REA 136
\$	J91(5).	J92(5).	J93(5).	J94(5).	J95(5).	REA 137
\$	J96(5).	J97(5).	J98(5).	J99(5).	J100(5)	REA 138
COMMON /TRANS2/	KKQ(100).	NNQ(100).	ISEQ(100).	NNKO.	I111(50)	REA 139
COMMON /TRANS4/	TL(1000).	OMEGAL(1000).	ASTAR(1000).	RSTAR(1000)		REA 140
COMMON /TRANV/	V1 (5).	V2 (5).	V3 (5).	V4 (5).	V5 (5).	REA 141
1	V6 (5).	V7 (5).	V8 (5).	V9 (5).	V10(5).	REA 142
2	V11(5).	V12(5).	V13(5).	V14(5).	V15(5).	REA 143
3	V16(5).	V17(5).	V18(5).	V19(5).	V20(5).	REA 144
4	V21(5).	V22(5).	V23(5).	V24(5).	V25(5).	REA 145
5	V26(5).	V27(5).	V28(5).	V29(5).	V30(5).	REA 146
6	V31(5).	V32(5).	V33(5).	V34(5).	V35(5).	REA 147
7	V36(5).	V37(5).	V38(5).	V39(5).	V40(5).	REA 148
8	V41(5).	V42(5).	V43(5).	V44(5).	V45(5).	REA 149
9	V46(5).	V47(5).	V48(5).	V49(5).	V50(5).	REA 150
\$	V51(5).	V52(5).	V53(5).	V54(5).	V55(5).	REA 151
\$	V56(5).	V57(5).	V58(5).	V59(5).	V60(5).	REA 152
\$	V61(5).	V62(5).	V63(5).	V64(5).	V65(5).	REA 153
\$	V66(5).	V67(5).	V68(5).	V69(5).	V70(5).	REA 154
\$	V71(5).	V72(5).	V73(5).	V74(5).	V75(5).	REA 155
\$	V76(5).	V77(5).	V78(5).	V79(5).	V80(5).	REA 156
\$	V81(5).	V82(5).	V83(5).	V84(5).	V85(5).	REA 157
\$	V86(5).	V87(5).	V88(5).	V89(5).	V90(5).	REA 158
\$	V91(5).	V92(5).	V93(5).	V94(5).	V95(5).	REA 159
\$	V96(5).	V97(5).	V98(5).	V99(5).	V100(5)	REA 160
1J(20).	SDGJ(20).	SHJAP(20).	THEVP(20)	CCJ(20).	DGJ(20).	SBREA 161
DIMENSION ASYM(RQ)						REA 162
DIMENSION CP(5,5)						REA 163
DIMENSION DUM(4)						REA 164
						REA 165

```

DIMENSION EPRP(2,10)
DIMENSION ELNAME(3),NATOM(3)
DIMENSION EPRP(2,10),SPRP(43,30),RPRP(29,92),GPRP(124,6),
1 ZPRP(64,10)
DIMENSION NSECTC(2)
DIMENSION SY1(3),SY2(3)
DIMENSION TNEP(186,2)
DIMENSION WORD(3,2)
DIMENSION WORD2(2)
EQUIVALENCE (AA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),
1 (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
2 (CAPX(1),GJ(1)),(SAJ(1),SBJ(1)),(SS(1),SDGJ(1)),
3 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
EQUIVALENCE (CP1(1),CP(1,1))
EQUIVALENCE (DUM(1),SUMGH)
EQUIVALENCE (EPI(1),EPRP(1,1))
EQUIVALENCE (EPI(1),EPRP(1,1))
EQUIVALENCE (GPI(1),GPRP(1,1))
EQUIVALENCE (NOZZLE,NPROFL(1))
EQUIVALENCE (RPI(1),RPRP(1,1))
EQUIVALENCE (SPI(1),SPRP(1,1))
EQUIVALENCE (TNEP(1,1),TNI(1))
EQUIVALENCE (ZPI(1),ZPRP(1,1))
DATA ASYM /1HH,2HHE,2HLI,2HBE,1HB,1HC,1HN,1HO,1HF,2HNE,2HNA,2HMG,
1 2HAL,2HSI,1HD,1HS,2HCL,2HAR,1HK,2HCA,2HSC,2HTI,1HV,2HCR,2HMN,
2 2HFE,2HCO,2HNI,2HCU,2HZN,2HGA,2HGE,2HAS,2HBR,2HHR,2HNB,
3 2HNR,1HY,2HZR,2HNB,2HMO,2HTC,2HRU,2HRH,2HPD,2HAG,2HCD,2HIN,
4 2HSN,2HSR,2HTE,1HI,2HXE,2HCE,2HBA,2HLA,2HCE,2HPR,2HND,2HPM,
5 2HSM,2HEU,2HGD,2HIB,2HDY,2HHC,2HER,2HTM,2HYB,2HLU,2HMF,2HTA,
6 1HW,2HRE,2HOS,2HIR,2HPT,2HAU,2HHG/
DATA IUPDI /1/
DATA PLUS /1H/, ONE /1/
DATA SHAPD /STRA,IGHT, LIN,E,CIRC,LE B,OTTO,M,
1 CIRC,LE T,OP, /
DATA SPNAME /SP1,SP2,SP3,SP4,SP5,SP6,SP7,SP8,
1 SP9,SP10,SP11,SP12,SP13,SP14,SP15,SP16,
2 SP17,SP18,SP19,SP20,SP21,SP22,SP23,SP24,
3 SP25,SP26,SP27,SP28,SP29,SP30/
DATA SY1,SY2 /3*1H(.3*1H)/
DATA WORD /6HTWO-DI,6HMENSIO,3HNAL,6H AXIS,6HYMMETR,2HIC/
DATA WORD2 /3HGAP,4HDIAM/
NAMELIST /INPUT/
*****GROUP 1 - GENERAL CONTROL VARIABLES
1 ISW1A,ISW2A,ISW3A,ISW4A,ISW6A,ISW1B,ISW3B,TWALL,NOTRAN,
1 STCPI,CXMAXI,READG,READXS,
*****GROUP 2 - OUTPUT CONTROLS
C ISW6H,ISW7B,TPRNTI,DATAP,NRECO,IRUN,
C *****GROUP 3 - RESERVOIR CONDITIONS
C ISW2B,PRESAI,FLOW,CIAP,HOSTAG,MFITER,
C *****GROUP 4 - GEOMETRY
4 NOZZLE,ICHAN,JDIM,NPRFLS,NPROFL,NBL,DIAM,NSECTS,ISHAPE,
4 PARAMI,ATPI,XZEROI,
*****GROUP 5 - GAS MODEL
C IGAS,AAMS,NCS,JCS,QPJ,ISCI,ISSI,ISRI,ICI,IE,IS,IR,ISATOM,
5 ISMOL,CTMXXI,BZEROI,INEOVI,

```

```

REA 166
REA 167
REA 168
REA 169
REA 170
REA 171
REA 172
REA 173
REA 174
REA 175
REA 176
REA 177
REA 178
REA 179
REA 180
REA 181
REA 182
REA 183
REA 184
REA 185
REA 186
REA 187
REA 188
REA 189
REA 190
REA 191
REA 192
REA 193
REA 194
REA 195
REA 196
REA 197
REA 198
REA 199
REA 200
REA 201
REA 202
REA 203
REA 204
REA 205
REA 206
REA 207
REA 208
REA 209
REA 210
REA 211
REA 212
REA 213
REA 214
REA 215
REA 216
REA 217
REA 218
REA 219
REA 220

```

ORIGINAL PAGE 17
OF POOR QUALITY

```

C *****GROUP 6 - TEST MODEL
6 XMODP1,NMODPT,TSDIAM,AXIMOD,KDIM, ,TAG,CATFAC,LEWIS,TMODEL,
6 TPLATE,
C *****GROUP 7 - WEDGE MODELS
7 ANGLE,ANGLE,NRADLE,RADLE,WX1,DWX,NWX,WX1,TWEDGE,WK,ISW9B,
C *****GROUP 8 - CONTROLS FOR THE FLOW SOLUTIONS
8 WSAVE,DELT11,DELT11,CCHI,NQSI,TTEST,GTEST,HTEST,TETEST,
8 QTEST,GAMIN,DCHELL,DCCHRT,
C *****GROUP 9 - ELECTRONIC NONEQUILIBRIUM
9 INT,KTF,KTR,ITR,EPAR,EPAR,TLIST,POM,
C *****GROUP 10 - CONTROL FOR DIAGNOSTIC DUMPS
$ ISW5A,ISW4B,ISW5B,ISW8B
$ NAMELIST /EINPUT/
1 EEP9,EEP10,NEELS,IEEP,SP1,SP2,SP3,SP4,SP5,EEP5,EEP6,EEP7,EEP8,
2 SP10,SP11,SP12,SP13,SP14,SP15,SP16,SP17,SP18,SP19,SP20,SP21,
3 SP22,SP23,SP24,SP25,SP26,SP27,SP28,SP29,SP30,RP1,RP2,RP3,RP4,
4 RP5,RP6,RP7,RP8,RP9,RP10,RP11,RP12,RP13,RP14,RP15,RP16,RP17,RP18,
5 RP19,RP20,RP21,RP22,RP23,RP24,RP25,RP26,RP27,RP28,RP29,RP30,
6 RP31,RP32,RP33,RP34,RP35,RP36,RP37,RP38,RP39,RP40,RP41,RP42,
7 RP43,RP44,RP45,RP46,RP47,RP48,RP49,RP50,RP51,RP52,RP53,RP54,
8 RP55,RP56,RP57,RP58,RP59,RP60,RP61,RP62,RP63,RP64,RP65,RP66,
9 RP67,RP68,RP69,RP70,RP71,RP72,RP73,RP74,RP75,RP76,RP77,RP78,
$ RP79,RP80,RP81,RP82,RP83,RP84,RP85,RP86,RP87,RP88,RP89,RP90,
$ RP91,RP92
$ NAMELIST /KQ,NQ,NKQ,ISEQ,V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,
1 V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,V21,V22,V23,V24,V25,V26,
2 V27,V28,V29,V30,V31,V32,V33,V34,V35,V36,V37,V38,V39,V40,V41,V42,
3 V43,V44,V45,V46,V47,V48,V49,V50,V51,V52,V53,V54,V55,V56,V57,V58,
4 V59,V60,V61,V62,V63,V64,V65,V66,V67,V68,V69,V70,V71,V72,V73,V74,
5 V75,V76,V77,V78,V79,V80,V81,V82,V83,V84,V85,V86,V87,V88,V89,V90,
6 V91,V92,V93,V94,V95,V96,V97,V98,V99,V100,V101,V102,V103,V104,
7 V105,V106,V107,V108,V109,V110,V111,V112,V113,V114,V115,V116,
8 V117,V118,V119,V120,V121,V122,V123,V124,
9 V125,V126,V127,V128,V129,V130,V131,V132,V133,V134,V135,V136,
0 V137,V138,V139,V140,V141,V142,V143,V144,V145,V146,V147,V148,
1 V149,V150,V151,V152,V153,V154,V155,V156,
2 V157,V158,V159,V160,V161,V162,V163,V164,V165,V166,V167,
3 V168,V169,V170,V171,V172,V173,V174,
4 V175,V176,V177,V178,V179,V180,V181,V182,V183,V184,V185,
5 V186,V187,
6 V188,V189,V190,V191,V192,V193,V194,V195,V196,V197,V198,
7 V199,V200,V201,V202,V203,V204,V205,V206,
8 V207,V208,V209,V210,V211,V212,V213,V214,V215,V216,
9 V217,V218,V219,V220,V221,V222,V223,V224,V225,V226,
0 V227,V228,V229,V230,V231,V232,V233,V234,V235,V236,
1 V237,V238,V239,V240,V241,V242,V243,V244,V245,V246,
2 V247,V248,V249,V250,V251,V252,V253,V254,V255,V256,
3 V257,V258,V259,V260,V261,V262,V263,V264,V265,V266,
4 V267,V268,V269,V270,V271,V272,V273,V274,V275,V276,
5 V277,V278,V279,V280,V281,V282,V283,V284,V285,V286,
6 V287,V288,V289,V290,V291,V292,V293,V294,V295,V296,
7 V297,V298,V299,V300,V301,V302,V303,V304,V305,V306,
8 V307,V308,V309,V310,V311,V312,V313,V314,V315,V316,
9 V317,V318,V319,V320,V321,V322,V323,V324,V325,V326,
0 V327,V328,V329,V330,V331,V332,V333,V334,V335,V336,
1 V337,V338,V339,V340,V341,V342,V343,V344,V345,V346,
2 V347,V348,V349,V350,V351,V352,V353,V354,V355,V356,
3 V357,V358,V359,V360,V361,V362,V363,V364,V365,V366,
4 V367,V368,V369,V370,V371,V372,V373,V374,V375,V376,
5 V377,V378,V379,V380,V381,V382,V383,V384,V385,V386,
6 V387,V388,V389,V390,V391,V392,V393,V394,V395,V396,
7 V397,V398,V399,V400,V401,V402,V403,V404,V405,V406,
8 V407,V408,V409,V410,V411,V412,V413,V414,V415,V416,
9 V417,V418,V419,V420,V421,V422,V423,V424,V425,V426,
0 V427,V428,V429,V430,V431,V432,V433,V434,V435,V436,
1 V437,V438,V439,V440,V441,V442,V443,V444,V445,V446,
2 V447,V448,V449,V450,V451,V452,V453,V454,V455,V456,
3 V457,V458,V459,V460,V461,V462,V463,V464,V465,V466,
4 V467,V468,V469,V470,V471,V472,V473,V474,V475,V476,
5 V477,V478,V479,V480,V481,V482,V483,V484,V485,V486,
6 V487,V488,V489,V490,V491,V492,V493,V494,V495,V496,
7 V497,V498,V499,V500,V501,V502,V503,V504,V505,V506,
8 V507,V508,V509,V510,V511,V512,V513,V514,V515,V516,
9 V517,V518,V519,V520,V521,V522,V523,V524,V525,V526,
0 V527,V528,V529,V530,V531,V532,V533,V534,V535,V536,
1 V537,V538,V539,V540,V541,V542,V543,V544,V545,V546,
2 V547,V548,V549,V550,V551,V552,V553,V554,V555,V556,
3 V557,V558,V559,V560,V561,V562,V563,V564,V565,V566,
4 V567,V568,V569,V570,V571,V572,V573,V574,V575,V576,
5 V577,V578,V579,V580,V581,V582,V583,V584,V585,V586,
6 V587,V588,V589,V590,V591,V592,V593,V594,V595,V596,
7 V597,V598,V599,V600,V601,V602,V603,V604,V605,V606,
8 V607,V608,V609,V610,V611,V612,V613,V614,V615,V616,
9 V617,V618,V619,V620,V621,V622,V623,V624,V625,V626,
0 V627,V628,V629,V630,V631,V632,V633,V634,V635,V636,
1 V637,V638,V639,V640,V641,V642,V643,V644,V645,V646,
2 V647,V648,V649,V650,V651,V652,V653,V654,V655,V656,
3 V657,V658,V659,V660,V661,V662,V663,V664,V665,V666,
4 V667,V668,V669,V670,V671,V672,V673,V674,V675,V676,
5 V677,V678,V679,V680,V681,V682,V683,V684,V685,V686,
6 V687,V688,V689,V690,V691,V692,V693,V694,V695,V696,
7 V697,V698,V699,V700,V701,V702,V703,V704,V705,V706,
8 V707,V708,V709,V710,V711,V712,V713,V714,V715,V716,
9 V717,V718,V719,V720,V721,V722,V723,V724,V725,V726,
0 V727,V728,V729,V730,V731,V732,V733,V734,V735,V736,
1 V737,V738,V739,V740,V741,V742,V743,V744,V745,V746,
2 V747,V748,V749,V750,V751,V752,V753,V754,V755,V756,
3 V757,V758,V759,V760,V761,V762,V763,V764,V765,V766,
4 V767,V768,V769,V770,V771,V772,V773,V774,V775,V776,
5 V777,V778,V779,V780,V781,V782,V783,V784,V785,V786,
6 V787,V788,V789,V790,V791,V792,V793,V794,V795,V796,
7 V797,V798,V799,V800,V801,V802,V803,V804,V805,V806,
8 V807,V808,V809,V810,V811,V812,V813,V814,V815,V816,
9 V817,V818,V819,V820,V821,V822,V823,V824,V825,V826,
0 V827,V828,V829,V830,V831,V832,V833,V834,V835,V836,
1 V837,V838,V839,V840,V841,V842,V843,V844,V845,V846,
2 V847,V848,V849,V850,V851,V852,V853,V854,V855,V856,
3 V857,V858,V859,V860,V861,V862,V863,V864,V865,V866,
4 V867,V868,V869,V870,V871,V872,V873,V874,V875,V876,
5 V877,V878,V879,V880,V881,V882,V883,V884,V885,V886,
6 V887,V888,V889,V890,V891,V892,V893,V894,V895,V896,
7 V897,V898,V899,V900,V901,V902,V903,V904,V905,V906,
8 V907,V908,V909,V910,V911,V912,V913,V914,V915,V916,
9 V917,V918,V919,V920,V921,V922,V923,V924,V925,V926,
0 V927,V928,V929,V930,V931,V932,V933,V934,V935,V936,
1 V937,V938,V939,V940,V941,V942,V943,V944,V945,V946,
2 V947,V948,V949,V950,V951,V952,V953,V954,V955,V956,
3 V957,V958,V959,V960,V961,V962,V963,V964,V965,V966,
4 V967,V968,V969,V970,V971,V972,V973,V974,V975,V976,
5 V977,V978,V979,V980,V981,V982,V983,V984,V985,V986,
6 V987,V988,V989,V990,V991,V992,V993,V994,V995,V996,
7 V997,V998,V999,V1000,V1001,V1002,V1003,V1004,V1005,
8 V1006,V1007,V1008,V1009,V1010,V1011,V1012,V1013,V1014,
9 V1015,V1016,V1017,V1018,V1019,V1020,V1021,V1022,V1023,
0 V1024,V1025,V1026,V1027,V1028,V1029,V1030,V1031,V1032,
1 V1033,V1034,V1035,V1036,V1037,V1038,V1039,V1040,V1041,
2 V1042,V1043,V1044,V1045,V1046,V1047,V1048,V1049,V1050,
3 V1051,V1052,V1053,V1054,V1055,V1056,V1057,V1058,V1059,
4 V1060,V1061,V1062,V1063,V1064,V1065,V1066,V1067,V1068,
5 V1069,V1070,V1071,V1072,V1073,V1074,V1075,V1076,V1077,
6 V1078,V1079,V1080,V1081,V1082,V1083,V1084,V1085,V1086,
7 V1087,V1088,V1089,V1090,V1091,V1092,V1093,V1094,V1095,
8 V1096,V1097,V1098,V1099,V1100,V1101,V1102,V1103,V1104,
9 V1105,V1106,V1107,V1108,V1109,V1110,V1111,V1112,V1113,
0 V1114,V1115,V1116,V1117,V1118,V1119,V1120,V1121,V1122,
1 V1123,V1124,V1125,V1126,V1127,V1128,V1129,V1130,V1131,
2 V1132,V1133,V1134,V1135,V1136,V1137,V1138,V1139,V1140,
3 V1141,V1142,V1143,V1144,V1145,V1146,V1147,V1148,V1149,
4 V1150,V1151,V1152,V1153,V1154,V1155,V1156,V1157,V1158,
5 V1159,V1160,V1161,V1162,V1163,V1164,V1165,V1166,V1167,
6 V1168,V1169,V1170,V1171,V1172,V1173,V1174,V1175,V1176,
7 V1177,V1178,V1179,V1180,V1181,V1182,V1183,V1184,V1185,
8 V1186,V1187,V1188,V1189,V1190,V1191,V1192,V1193,V1194,
9 V1195,V1196,V1197,V1198,V1199,V1200,V1201,V1202,V1203,
0 V1204,V1205,V1206,V1207,V1208,V1209,V1210,V1211,V1212,
1 V1213,V1214,V1215,V1216,V1217,V1218,V1219,V1220,V1221,
2 V1222,V1223,V1224,V1225,V1226,V1227,V1228,V1229,V1230,
3 V1231,V1232,V1233,V1234,V1235,V1236,V1237,V1238,V1239,
4 V1240,V1241,V1242,V1243,V1244,V1245,V1246,V1247,V1248,
5 V1249,V1250,V1251,V1252,V1253,V1254,V1255,V1256,V1257,
6 V1258,V1259,V1260,V1261,V1262,V1263,V1264,V1265,V1266,
7 V1267,V1268,V1269,V1270,V1271,V1272,V1273,V1274,V1275,
8 V1276,V1277,V1278,V1279,V1280,V1281,V1282,V1283,V1284,
9 V1285,V1286,V1287,V1288,V1289,V1290,V1291,V1292,V1293,
0 V1294,V1295,V1296,V1297,V1298,V1299,V1300,V1301,V1302,
1 V1303,V1304,V1305,V1306,V1307,V1308,V1309,V1310,V1311,
2 V1312,V1313,V1314,V1315,V1316,V1317,V1318,V1319,V1320,
3 V1321,V1322,V1323,V1324,V1325,V1326,V1327,V1328,V1329,
4 V1330,V1331,V1332,V1333,V1334,V1335,V1336,V1337,V1338,
5 V1339,V1340,V1341,V1342,V1343,V1344,V1345,V1346,V1347,
6 V1348,V1349,V1350,V1351,V1352,V1353,V1354,V1355,V1356,
7 V1357,V1358,V1359,V1360,V1361,V1362,V1363,V1364,V1365,
8 V1366,V1367,V1368,V1369,V1370,V1371,V1372,V1373,V1374,
9 V1375,V1376,V1377,V1378,V1379,V1380,V1381,V1382,V1383,
0 V1384,V1385,V1386,V1387,V1388,V1389,V1390,V1391,V1392,
1 V1393,V1394,V1395,V1396,V1397,V1398,V1399,V1400,V1401,
2 V1402,V1403,V1404,V1405,V1406,V1407,V1408,V1409,V1410,
3 V1411,V1412,V1413,V1414,V1415,V1416,V1417,V1418,V1419,
4 V1420,V1421,V1422,V1423,V1424,V1425,V1426,V1427,V1428,
5 V1429,V1430,V1431,V1432,V1433,V1434,V1435,V1436,V1437,
6 V1438,V1439,V1440,V1441,V1442,V1443,V1444,V1445,V1446,
7 V1447,V1448,V1449,V1450,V1451,V1452,V1453,V1454,V1455,
8 V1456,V1457,V1458,V1459,V1460,V1461,V1462,V1463,V1464,
9 V1465,V1466,V1467,V1468,V1469,V1470,V1471,V1472,V1473,
0 V1474,V1475,V1476,V1477,V1478,V1479,V1480,V1481,V1482,
1 V1483,V1484,V1485,V1486,V1487,V1488,V1489,V1490,V1491,
2 V1492,V1493,V1494,V1495,V1496,V1497,V1498,V1499,V1500,
3 V1501,V1502,V1503,V1504,V1505,V1506,V1507,V1508,V1509,
4 V1510,V1511,V1512,V1513,V1514,V1515,V1516,V1517,V1518,
5 V1519,V1520,V1521,V1522,V1523,V1524,V1525,V1526,V1527,
6 V1528,V1529,V1530,V1531,V1532,V1533,V1534,V1535,V1536,
7 V1537,V1538,V1539,V1540,V1541,V1542,V1543,V1544,V1545,
8 V1546,V1547,V1548,V1549,V1550,V1551,V1552,V1553,V1554,
9 V1555,V1556,V1557,V1558,V1559,V1560,V1561,V1562,V1563,
0 V1564,V1565,V1566,V1567,V1568,V1569,V1570,V1571,V1572,
1 V1573,V1574,V1575,V1576,V1577,V1578,V1579,V1580,V1581,
2 V1582,V1583,V1584,V1585,V1586,V1587,V1588,V1589,V1590,
3 V1591,V1592,V1593,V1594,V1595,V1596,V1597,V1598,V1599,
4 V1600,V1601,V1602,V1603,V1604,V1605,V1606,V1607,V1608,
5 V1609,V1610,V1611,V1612,V1613,V1614,V1615,V1616,V1617,
6 V1618,V1619,V1620,V1621,V1622,V1623,V1624,V1625,V1626,
7 V1627,V1628,V1629,V1630,V1631,V1632,V1633,V1634,V1635,
8 V1636,V1637,V1638,V1639,V1640,V1641,V1642,V1643,V1644,
9 V1645,V1646,V1647,V1648,V1649,V1650,V1651,V1652,V1653,
0 V1654,V1655,V1656,V1657,V1658,V1659,V1660,V1661,V1662,
1 V1663,V1664,V1665,V1666,V1667,V1668,V1669,V1670,V1671,
2 V1672,V1673,V1674,V1675,V1676,V1677,V1678,V1679,V1680,
3 V1681,V1682,V1683,V1684,V1685,V1686,V1687,V1688,V1689,
4 V1690,V1691,V1692,V1693,V1694,V1695,V1696,V1697,V1698,
5 V1699,V1700,V1701,V1702,V1703,V1704,V1705,V1706,V1707,
6 V1708,V1709,V1710,V1711,V1712,V1713,V1714,V1715,V1716,
7 V1717,V1718,V1719,V1720,V1721,V1722,V1723,V1724,V1725,
8 V1726,V1727,V1728,V1729,V1730,V1731,V1732,V1733,V1734,
9 V1735,V1736,V1737,V1738,V1739,V1740,V1741,V1742,V1743,
0 V1744,V1745,V1746,V1747,V1748,V1749,V1750,V1751,V1752,
1 V1753,V1754,V1755,V1756,V1757,V1758,V1759,V1760,V1761,
2 V1762,V1763,V1764,V1765,V1766,V1767,V1768,V1769,V1770,
3 V1771,V1772,V1773,V1774,V1775,V1776,V1777,V1778,V1779,
4 V1780,V1781,V1782,V1783,V1784,V1785,V1786,V1787,V1788,
5 V1789,V1790,V1791,V1792,V1793,V1794,V1795,V1796,V1797,
6 V1798,V1799,V1800,V1801,V1802,V1803,V1804,V1805,V1806,
7 V1807,V1808,V1809,V1810,V1811,V1812,V1813,V1814,V1815,
8 V1816,V1817,V1818,V1819,V1820,V1821,V1822,V1823,V1824,
9 V1825,V1826,V1827,V1828,V1829,V1830,V1831,V1832,V1833,
0 V1834,V1835,V1836,V1837,V1838,V1839,V1840,V1841,V1842,
1 V1843,V1844,V1845,V1846,V1847,V1848,V1849,V1850,V1851,
2 V1852,V1853,V1854,V1855,V1856,V1857,V1858,V1859,V1860,
3 V1861,V1862,V1863,V1864,V1865,V1866,V1867,V1868,V1869,
4 V1870,V1871,V1872,V1873,V1874,V1875,V1876,V1877,V1878,
5 V1879,V1880,V1881,V1882,V1883,V1884,V1885,V1886,V1887,
6 V1888,V1889,V1890,V1891,V1892,V1893,V1894,V1895,V1896,
7 V1897,V1898,V1899,V1900,V1901,V1902,V1903,V1904,V1905,
8 V1906,V1907,V1908,V1909,V1910,V1911,V1912,V1913,V1914,
9 V1915,V1916,V1917,V1918,V1919,V1920,V1921,V1922,V1923,
0 V1924,V1925,V1926,V1927,V1928,V1929,V1930,V1931,V1932,
1 V1933,V1934,V1935,V1936,V1937,V1938,V1939,V1940,V1941,
2 V1942,V1943,V1944,V1945,V1946,V1947,V1948,V1949,V1950,
3 V1951,V1952,V1953,V1954,V1955,V1956,V1957,V1958,V1959,
4 V1960,V1961,V1962,V1963,V1964,V1965,V1966,V1967,V1968,
5 V1969,V1970,V1971,V1972,V1973,V1974,V1975,V1976,V1977,
6 V1978,V1979,V1980,V1981,V1982,V1983,V1984,V1985,V1986,
7 V1987,V1988,V1989,V1990,V1991,V1992,V1993,V1994,V1995,
8 V1996,V1997,V1998,V1999,V2000,V2001,V2002,V2003,V2004,
9 V2005,V2006,V2007,V2008,V2009,V2010,V2011,V2012,V2013,
0 V2014,V2015,V2016,V2017,V2018,V2019,V2020,V2021,V2022,
1 V2023,V2024,V2025,V2026,V2027,V2028,V2029,V2030,V2031,
2 V2032,V2033,V2034,V2035,V2036,V2037,V2038,V2039,V2040,
3 V2041,V2042,V2043,V2044,V2045,V2046,V2047,V2048,V2049,
4 V2050,V2051,V2052,V2053,V2054,V2055,V2056,V2057,V2058,
5 V2059,V2060,V2061,V2062,V2063,V2064,V2065,V2066,V2067,
6 V2068,V2069,V2070,V2071,V2072,V2073,V2074,V2075,V2076,
7 V2077,V2078,V2079,V2080,V2081,V2082,V2083,V2084,V2085,
8 V2086,V2087,V2088,V2089,V2090,V2091,V2092,V2093,V2094,
9 V2095,V2096,V2097,V2098,V2099,V2100,V2101,V2102,V2103,
0 V2104,V2105,V2106,V2107,V2108,V2109,V2110,V2111,V2112,
1 V2113,V2114,V2115,V2116,V2117,V2118,V2119,V2120,V2121,
2 V2122,V2123,V2124,V2125,V2126,V2127,V2128,V2129,V2130,
3 V2131,V2132,V2133,V2134,V2135,V2136,V2137,V2138,V2139,
4 V2140,V2141,V2142,V2143,V2144,V2145,V2146,V2147,V2148,
5 V2149,V2150,V2151,V2152,V2153,V2154,V2155,V2156,V2157,
6 V2158,V2159,V2160,V2161,V2162,V2163,V2164,V2165,V2166,
7 V2167,V2168,V2169,V2170,V2171,V2172,V2173,V2174,V2175,
8 V2176,V2177,V2178,V2179,V2180,V2181,V2182,V2183,V2184,
9 V2185,V2186,V2187,V2188,V2189,V2190,V2191,V2192,V2193,
0 V2194,V2195,V2196,V2197,V2198,V2199,V2200,V2201,V2202,
1 V2203,V2204,V2205,V2206,V2207,V2208,V2209,V2210,V2211,
2 V2212,V2213,V2214,V2215,V2216,V2217,V2218,V2219,V2220,
3 V2221,V2222,V2223,V2224,V2225,V2226,V2227,V2228,V2229,
4 V2230,V2231,V2232,V2233,V2234,V2235,V2236,V2237,V2238,
5 V2239,V2240,V2241,V2242,V2243,V2244,V2245,V2246,V2247,
6 V2248,V2249,V2250,V2251,V2252,V2253,V2254,V2255,V2256,
7 V2257,V2258,V2259,V2260,V2261,V2262,V2263,V2264,V2265,
8 V2266,V2267,V2268,V2269,V2270,V2271,V2272,V2273,V2274,
9 V2275,V2276,V2277,V2278,V2279,V2280,V2281,V2282,V2283,
0 V2284,V2285,V2286,V2287,V2288,V2289,V2290,V2291,V2292,
1 V2293,V2294,V2295,V2296,V2297,V2298,V2299,V2300,V2301,
2 V2302,V2303,V2304,V2305,V2306,V2307,V2308,V2309,V2310,
3 V2311,V2312,V2313,V2314,V2315,V2316,V2317,V2318,V2319,
4 V2320,V2321,V2322,V2323,V2324,V2325,V2326,V2327,V2328,
5 V2329,V2330,V2331,V2332,V2333,V2334,V2335,V2336,V2337,
6 V2338,V2339,V2340,V2341,V2342,V2343,V2344,V2345,V2346,
7 V2347,V2348,V2349,V2350,V2351,V2352,V2353,V2354,V2355,
8 V2356,V2357,V2358,V2359,V2360,V2361,V2362,V2363,V2364,
9 V2365,V2366,V2367,V2368,V2369,V2370,V2371,V2372,V2373,
0 V2374,V2375,V2376,V2377,V2378,V2379,V2380,V2381,V2382,
1 V2383,V2384,V2385,V2386,V2387,V2388,V2389,V2390,V2391
```

```

10 SPRP(1,1)=SPNAME(1)
20 CONTINUE
   IF (.NOT. READ(5)) GO TO 30
   READ(5, TINPUT)
   WRITE(6, TINPUT)
30 CONTINUE
   ICASE=ICASE+1
   WRITE(6, 980) IRUN, ICASE, (ACOM(I), I=1, 20)
   LETIGS=AAMS, AND, (IGAS, EQ, 1, OR, IGAS, EQ, 2)
   DO 40 I=1, 4
   DUM(I)=0.
   IF (ICHAN, GT, 0) NPRFLS=2
   IF (NPRFLS, EQ, 1, OR, NPRFLS, EQ, 2) GO TO 50
   WRITE(6, 990) NPRFLS
50 STOP
   DO 60 I=1, NPRFLS
   IF (ICHAN, GT, 0) NPROFL(I)=CP(I, ICHAN)
   IPRFL=NPROFL(I)
   IF (IPRFL, GT, 0) DIAM(I)=ZPRP(1, IPRFL)/1.27
60 CONTINUE
   AXISYM=AXIMOD
   IF (NPRFLS, EQ, 2) AXISYM=.FALSE.
   IF (ISW2B) 70, 80, 170
   HS=HSTAG/1.8
   WRITE(6, 1000) HSTAG, FLOW
   GO TO 90
80 WRITE(6, 1010) PRESAT, FLOW
90 IF (NPRFLS, EQ, 2) GO TO 110
   IF (JDIM, NE, 0) GO TO 100
   SMASS=70.3069*FLOW/DIAM(1)
   GO TO 120
100 SMASS=89.5173*FLOW/(DIAM(1))**.2
   GO TO 120
110 SMASS=70.3069*FLOW/DIAM(1)/DIAM(2)
120 IF (.NOT. SETIGS) GO TO 180
   IF (ISW2B, EQ, 0) GO TO 130
   HWINOV=HSTAG
   GO TO 14
130 HWINOV=(-S, 7*PRESAT/SMASS)**2.52
140 IF (HWINOV-8000.) 150, 150, 160
150 IGAS=2
   GO TO 180
160 IGAS=1
   GO TO 180
170 WRITE(6, 1020) PRESAT, CTAPI
   IF (.NOT. SETIGS) GO TO 180
   IF (CTAPI-6000.) 150, 150, 160
180 IUPD=IUPDI
   NQS=NQSI
   IF (FSTAG, NE, 0.) FSTAG=SIGN(ONE, FSTAG)
   RSA=1.
   IF (NPRFLS, EQ, 2) GO TO 190
   NBL=1
   GO TO 200

```

ORIGINAL PAGE IS
OF POOR QUALITY

REA 275
 REA 276
 REA 277
 REA 278
 REA 279
 REA 280
 REA 281
 REA 282
 REA 283
 REA 284
 REA 285
 REA 286
 REA 287
 REA 288
 REA 289
 REA 290
 REA 291
 REA 292
 REA 293
 REA 294
 REA 295
 REA 296
 REA 297
 REA 298
 REA 299
 REA 300
 REA 301
 REA 302
 REA 303
 REA 304
 REA 305
 REA 306
 REA 307
 REA 308
 REA 309
 REA 310
 REA 311
 REA 312
 REA 313
 REA 314
 REA 315
 REA 316
 REA 317
 REA 318
 REA 319
 REA 320
 REA 321
 REA 322
 REA 323
 REA 324
 REA 325
 REA 326
 REA 327
 REA 328
 REA 329

ORIGINAL PAGE IS
OF POOR QUALITY

```

190 IF (ICHAN.GT.0) NBL=CP(4,ICHAN)
200 IF (ISW7B.NF.0) ISW7B=1
    LIMOUT=ISW7B*ISW3B
    INEQV=INEQVI
    PRESA=PRESAI
    CTAP=CTAPI
    CTMAX=CTMXXI
    CXMAX=CXMAXI*2.5
    MBL=1
    IF (NPRFLS.EQ.1) GO TO 210
    IF (NHL.EQ.1) MBL=2
    NTS=0
    DO 230 I=1,20
    IF (TSDIAM(I).LE.DIAM(MBL)) TSDIAM(I)=1.E30
    IF (TSDIAM(I).GT.9.E19) GO TO 230
    NTS=NTS+1
    IF (NPRFLS.EQ.2) GO TO 220
    TSAR(NTS)=(TSDIAM(I)/DIAM(1))*((JD(I)+1)
    GO TO 230
    TSAR(NTS)=1.27*TSDIAM(1)
    CONTINUE
    IF (NANGLE.EQ.0.OR.NRADLE.EQ.0) GO TO 240
    IF (NX.EQ.0.AND.WXI(1).GT.1.E19) GO TO 240
    IF (NPRFLS.EQ.2) GO TO 240
    WEDGEM=.TRUE.
    GO TO 250
    WEDGEM=.FALSE.
    IF (NOZZLE.GT.0) GO TO 260
    XZERO=2.54*XZEROI
    GO TO 270
    XZERO=ZPRP(2,NOZZLE)
    IN THE CASE OF A CHANNEL, THE CODE USES THE XZERO FOR THE FIRST
    PROFILE
    SL=1.
    XZERO=BZEROI
    TSTOP=TSTOPT
    TPRINT=TPRINTI
    DELTAX=DELTXI
    DELTI=DELTII
    DO 320 J=1,NPRFLS
    IPRFL=NPROFL(J)
    IF (IPRFL.EQ.0) GO TO 300
    RTHCM(J)=ZPRP(1,IPRFL)
    NSECTU(J)=ZPRP(3,IPRFL)
    NSECTD(J)=ZPRP(4,IPRFL)
    NSECT(J)=NSECTU(J)+NSECTD(J)
    IF (J.EQ.1) FACNAM=ZPRP(64,IPRFL)
    NSECTJ=NSECT(J)
    DO 290 I=1,NSECTJ
    ISHAPE(I,J)=ZPRP(I+4,IPRFL)
    IF (I.LT.NSECTJ) ATPI(I,J)=ZPRP(I+16,IPRFL)
    DO 280 L=1,3
    LL=3*I+L+24
    PARAM(L,I,J)=ZPRP(LL,IPRFL)
    CONTINUE
230
290

```

```

300 GO TO 320
    NSECTU(J)=NSECTS(1,J)
    NSECTD(J)=NSECTS(2,J)
    NSECT(J)=NSECTU(J)+NSECTD(J)
    IF (J.EQ.1) FACNAM=ACOM(1)
    RTHCM(J)=1,27*DIAM(J)
    NSECTJ=NSECT(J)
    DO 310 I=1,NSECTJ
    DO 310 L=1,3
    PARAM(L,I,J)=PARAMI(L,I,J)
    CONTINUE
    IF (ICHAN.EQ.0) GO TO 330
    CHANAM=CP(3,ICHAN)
    FACNAM=CP(5,ICHAN)
    GO TO 340
    CHANAM=ACOM(2)
    CONTINUE
    IF (NPRFLS.EQ.1) GO TO 360
    IF (ICHAN.NE.0) GO TO 350
    WRITE (6,1030) CHANAM,FACNAM,DIAM(1),DIAM(2)
    GO TO 390
    WRITE (6,1040) CHANAM,ICHAN,FACNAM,DIAM(1),DIAM(2)
    GO TO 390
    JDPI=JDIM+1
    IF (NOZZLE.GT.0) GO TO 370
    WRITE (6,1050) (WORD(I,JDPI),I=1,3),DIAM(1),WORD2(JDPI),FACNAM
    GO TO 390
    WRITE (6,1060) (WORD(I,JDPI),I=1,3),NOZZLE,DIAM(1),WORD2(JDPI),FACNAM
    1NAV
    IF (NOZZLE.EQ.NOZZO.AND.NOZZLE.NE.0) GO TO 440
    GO TO 400
    IF (ICHAN.EQ.ICHANO.AND.ICHAN.NE.0) GO TO 440
    DO 430 J=1,NPRFLS
    IF (NPRFLS.EQ.1) GO TO 410
    WRITE (6,1070) J
    WRITE (6,1080) RTHCM(J),XZERO,NSECT(J),NSECTU(J)
    WRITE (6,1090)
    NSM1=NSECT(J)-1
    DO 420 K=1,NSM1
    ISH=ISHAPE(K,J)
    WRITE (6,1100) K,ISH,(SHAPD(I,ISH),I=1,4),ATPI(K,J),(PARAM(L,K,J),
    1L=1,3)
    NSECTJ=NSECT(J)
    K=NSECTJ
    ISH=ISHAPE(NSECTJ,J)
    WRITE (6,1110) NSECTJ,ISH,(SHAPD(I,ISH),I=1,4),(PARAM(L,K,J),L=1,3)
    1) CONTINUE
    CONTINUE
    IF (NEELS.EQ.0) GO TO 460
    DO 450 I=1,NEELS
    IZ=EEPRP(1,I)+0.1
    II=IEEP(I)
    EPRP(1,II)=ASYM(IZ)
    EPRP(2,II)=EEPRP(2,I)
450
430 CONTINUE
440

```

```

460 IGS=IGAS
    IF (IGAS.NE.0) GO TO 470
    WRITE (6,1120)
    ISS=ISCI
    ISS=ISSI
    ISR=ISRI
    IC=ICI
    GO TO 520
470 IGAS=IGAS(IGAS)
    WRITE (6,1130) 'GAS,GPRP(1,IGAS)
    SET UP DESCRIPTION OF GAS IN TERMS OF ELEMENTS, SPECIES, AND
    C REACTIONS
    ISC=GPRP(2,IGAS)
    ISS=GPRP(3,IGAS)
    ISR=GPRP(4,IGAS)
    IC=GPRP(5,IGAS)
    DO 480 I=1,ISC
    IE(I)=GPRP(I+5,IGAS)
    NCS=GPRP(120,IGAS)
    DO 490 I=1,NCS
    IF (IGS.LT.0) GO TO 490
    OPJ(I)=GPRP(I+15,IGAS)
    JCS(I)=GPRP(I+109,IGAS)
    DO 500 I=1,ISS
    IS(I)=GPRP(I+25,IGAS)
    DO 510 I=1,ISR
    IR(I)=GPRP(I+45,IGAS)
    ISATOM=GPRP(121,IGAS)
    ISMOL=GPRP(122,IGAS)
    INT=GPRP(123,IGAS)
    LEWIS=GPRP(124,IGAS)
    IF (INT.EQ.0) GO TO 530
    NT=2
    GO TO 540
530 NT=1
540 CONTINUE
    LOOK UP ELEMENTS
    DO 550 I=1,ISC
    II=IE(I)
    ELEMENT(I)=EPRP(1,II)
    CMW(I)=EPRP(2,II)
    LOOK UP SPECIES
    IAMBIP=1
    NFIT=0
    DO 560 I=1,ISS
    II=IS(I)
    DO 560 K=1,NCS
    IF (II.EQ.JCS(K)) GO TO 570
    CONTINUE
    GO TO 580
560 IJCS(K)=I
    CONTINUE
    IF (II.EQ.ISATOM) JATOM=I
    IF (II.EQ.ISMOL) JMOL=I
    HP(I)=SPRP(1,II)

```

```

REA 440
REA 441
REA 442
REA 443
REA 444
REA 445
REA 446
REA 447
REA 448
REA 449
REA 450
REA 451
REA 452
REA 453
REA 454
REA 455
REA 456
REA 457
REA 458
REA 459
REA 460
REA 461
REA 462
REA 463
REA 464
REA 465
REA 466
REA 467
REA 468
REA 469
REA 470
REA 471
REA 472
REA 473
REA 474
REA 475
REA 476
REA 477
REA 478
REA 479
REA 480
REA 481
REA 482
REA 483
REA 484
REA 485
REA 486
REA 487
REA 488
REA 489
REA 490
REA 491
REA 492
REA 493
REA 494

```

```

NELS=SPRP(2,II)+0.1
DO 630 K=1,ISC
DO 590 J=1,NELS
IES=SPRP(J+2,II)+0.1
IF (IE(K).EQ.IES) GO TO 600
CONTINUE
LPIJ(I,K)=0
GO TO 630
AL=SPRP(J+5,II)
LP=AL
DAL=AL-LP
IF (ABS(DAL).LE.0.5) GO TO 620
IF (DAL.GT.0.) GO TO 610
LP=LP-1
GO TO 620
LP=LP+1
LPIJ(I,K)=LP
IF (II.EQ.ISATOM.AND.IES.EQ.1) IAMBIP=2
CONTINUE
TFA(I)=SPRP(9,II)
TFB(I)=SPRP(10,II)
TFC(I)=SPRP(11,II)
TFD(I)=SPRP(12,II)
TFE(I)=SPRP(13,II)
TFK(I)=SPRP(14,II)
SHJAP(I)=SPRP(15,II)
ETAJ(I)=SPRP(16,II)
SBJ(I)=SPRP(17,II)
THEVP(I)=SPRP(18,II)
IF (ETAJ(I).LT.2.9) GO TO 650
DO 640 K=1,3
THEVE(K+1,I)=SPRP(K+40,II)
IGM(I)=SPRP(19,II)+0.1
IGJ(I)=SPRP(20,II)+0.1
IF (IGJ(I).NE.0) NFIT=1
IF (ETAJ(I).NE.0.OR.IGJ(I).NE.0) GO TO 660
WRITE (6,II90) I,II
STOP
IGMV=IGM(I)
DO 670 L=1,IGMV
GELJ(L,I)=SPRP(L+20,II)
ELJ(L,I)=SPRP(L+30,II)
CONTINUE
DO 700 I=1,ISC
CAPQ(I)=0.
DO 700 J=1,NCS
JJ=JCS(J)
NELS=SPRP(2,JJ)+0.1
DO 690 K=1,NELS
KK=SPRP(K+2,JJ)+0.1
IF (KK.NE.IE(I)) GO TO 690
CAPQ(I)=CAPQ(I)+QPJ(J)*SPRP(K+3,JJ)
CONTINUE
LOOK UP REACTIONS

```

```

DO 780 I=1,ISR
II=IR(I)
CAI(I)=RPRP(1,II)
ETAI(I)=RPRP(2,II)
CEACT(I)=RPRP(3,II)
QO(I)=RPRP(4,II)
NPSS=RPRP(5,II)+0.1
NPSS=RPRP(6,II)+0.1
DO 770 J=1,ISS
DO 710 K=1,NR3S
JRS=RPRP(K+6,II)+0.1
IF (JRS.NE.IS(J)) GO TO 710
XNUIJ(I,J)=RPRP(K+12,II)
GO TO 720
CONTINUE
XNUIJ(I,J)=0.
CONTINUE
DO 730 K=1,NPSS
JPS=RPRP(K+9,II)+0.1
IF (JPS.NE.IS(J)) GO TO 730
XNUIJP(I,J)=RPRP(K+15,II)
GO TO 740
CONTINUE
XNUIJP(I,J)=0.
CONTINUE
NK=RPRP(19,II)+0.1
DO 750 K=1,NK
ISKUR=RPRP(K+19,II)+0.1
IF (ISKUR.EQ.IS(J)) GO TO 760
CONTINUE
KUR(I,J)=0
GO TO 770
KUR(I,J)=1
CONTINUE
IF (IGAS.EQ.0.OR.NT.EQ.1) GO TO 780
KTF(I)=TNEP(I,INT)
KTR(I)=TNEP(I+25,INT)
ITR(I)=TNEP(I+50,INT)
EPAR(1,I)=TNEP(I+75,INT)
EPAR(2,I)=TNEP(I+100,INT)
CONTINUE
IF (NT.EQ.1) GO TO 860
IF (IGAS.EQ.0) GO TO 800
DO 790 I=1,30
TLIST(I)=TNEP(I+125,INT)
POM(I)=TNEP(I+155,INT)
BPAR=TNEP(186,INT)
DO 820 I=2,ISS
IF (LPIJ(I,1).EQ.0) GO TO 810
ICH(I)=2
GO TO 820
ICH(I)=1
CONTINUE
DO 850 I=1,ISR
IF (KTF(I).NE.4) GO TO 840

```

```

REA 550
REA 551
REA 552
REA 553
REA 554
REA 555
REA 556
REA 557
REA 558
REA 559
REA 560
REA 561
REA 562
REA 563
REA 564
REA 565
REA 566
REA 567
REA 568
REA 569
REA 570
REA 571
REA 572
REA 573
REA 574
REA 575
REA 576
REA 577
REA 578
REA 579
REA 580
REA 581
REA 582
REA 583
REA 584
REA 585
REA 586
REA 587
REA 588
REA 589
REA 590
REA 591
REA 592
REA 593
REA 594
REA 595
REA 596
REA 597
REA 598
REA 599
REA 600
REA 601
REA 602
REA 603
REA 604

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
830 DO 830 J=2,ISS
840 IF (XNU1JP(I,J).EQ.0.) GO TO 830
850 IPA(I)=J
860 CONTINUE
      GO TO 850
870 CONTINUE
      IPA(I)=0
      CONTINUE
      IZERO=0
      SUPGO=IGAS.NE.0.AND.IGAS.EQ.IGASD
      IF (SUPGO) GO TO 870
      WRITE (6,1140)
      WRITE (6,1150)
      CGMW=C.
      DO 890 I=1,NCS
      JJ=JCS(I)
      NELS=SPRP(2,JJ)
      CSMW=0.
      DO 880 K=1,NELS
      KK=SPRP(K+2,JJ)
      ELNAME(K)=EPRP(1,KK)
      NATOM(K)=SPRP(K+5,JJ)
      CSMW=CSMW+NATOM(K)*EPRP(2,KK)
      IF (KK.NE.1) GO TO 880
      ELNAME(K)=PLUS
      NATOM(K)=--NATOM(K)
      CONTINUE
      CGMW=CGMW+CSMW*OPJ(I)
      IF (SUPGO) GO TO 890
      WRITE (6,1160) I,SPRP(1,JJ),JJ,OPJ(I),CSMW,(SY1(K),ELNAME(K),SY2(K)
      1),NATOM(K),K=1,NELS)
      CONTINUE
      IF (.NOT.SUPGO) WRITE (6,1170) CGMW
      IGASO=IGAS
      NOZZO=NOZZLE
      ICHANQ=ICHAN
      IGAS=IGS
      IF (XMODP1) 900,900,910
      XMODP1=1.E20
      WRITE (6,1180)
      XMP1=XMODP1*2.54
      IF (CXMAXI.GT.XMODP1.AND.NMODPT.GT.1) GO TO 920
      FACMP=1.E10
      GO TO 930
      FNMP=NMODPT-1
      EXPMP=1./FNMP
      FACMP=(CXMAXI/XMODP1)**EXPMP
      CONTINUE
      IF (NRECO.LE.0.OR.ICASE.GT.1) GO TO 950
      DO 940 I=1,NRECO
      READ (ITPOUT)
      CONTINUE
      RETURN
      C
      C
      C
```

```

960 960 660
970 REA 661
980 REA 662
    1.12H IN THIS JOB.9X.20A4///)
990 REA 663
1000 REA 664
    FORMAT (30H0 INVALID INPUT DATA... NPRFLS=.110)
1010 REA 665
    FORMAT (21H STAGNATION ENTHALPY=.F8.0.25H BTU/LB, TOTAL MASS FLOW=.F9.5.7H LB/SEC)
1020 REA 666
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.23H ATM, TOTAL MASS FLOW=.F9.5.7H LB/SEC)
1030 REA 667
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1040 REA 668
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1050 REA 669
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1060 REA 670
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1070 REA 671
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1080 REA 672
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1090 REA 673
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1100 REA 674
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1110 REA 675
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1120 REA 676
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1130 REA 677
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1140 REA 678
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1150 REA 679
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1160 REA 680
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1170 REA 681
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1180 REA 682
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
1190 REA 683
    FORMAT (20H RESERVOIR PRESSURE=.F9.4.28H ATM, RESERVOIR TEMPERATURE=.F9.4.28H ATM)
    END

```

**SUBROUTINE RESET
SIMULATION OF UNIVAC 1108 TIMING ROUTINES RESET AND TIME**

1 RST
2 RST
3 RST
4 RST
5 RST
6 RST
7 RST
8 RST
9 RST
10 RST
11 RST
12 RST
13- RST

C
10
20
30
40

```

SUBROUTINE RESET
IN=1
GO TO 10
ENTRY TIME(I)
IN=2
CALL ACUCPU (ICPU)
GO TO (20,30), IN
IZERO=ICPU
GO TO 40
I=IZERC-ICPU
RETURN
END
    
```

**ORIGINAL PAGE 11
OF POOR QUALITY**


```

1 SUBROUTINE RESTMP
2 CALCULATES RESERVOIR TEMPERATURE FROM MASS FLOW DATA.
3 DOUBLE PRECISION AM,BM
4 LOGICAL ERR,SUPGO
5 REAL ACCM(30),ELMENT(10),HP(20)
6 DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
7 COMMON AA(22,24),CDIJ(20,10),CAPX(20)
8 COMMON A,AFTS,AFNX,AMACH,
9 C,CARB,CH,CST,CT,CTAP,
10 CRRB,CRS,CSTP,CTP,CTPL,CX,CXB,CXMAX,DATEST,DBTEST,DELTA,
11 DELT2,DELTAX,DLOGA,DLOGR,DT,DLOGR,PRESTH,PRHC,RHAP,
12 DPCTEST,PRES,PRESB,RHOP,RHOC,RHPL,RHTR,ROBARP,SCPG,SL,SL64,
13 RHOBAR,RHUC,RHOP,SC,SL,SL64,
14 SEN,SHPG,SC,SL,SL64,
15 TEST,YESTB,TPRINT,TSTOP,UP,
16 COMMON RE(64),HET(20),BLBK(31),CAI(64),CAPXTH(20),
17 CCRPJ(20),CEACT(64),CGI(20),CGMU(20),CHI(64),CHI(20),
18 CLNINC(64),CLNPI(64),CMW(20),ETAI(64),ETAJ(20),GJA(20),
19 GJ3(20),PERTGJ(20),PGJ(20),PI(64),PICHI(64),QM(20),
20 QQ(64),SAJ(20),SDCHI(64),SENT(20),SHJ(20),SHJA(20),
21 SKIL(20),SS(20),TB(30),TEA(20),TFB(20),TFC(20),
22 TFD(20),TFE(20),TFK(20),THEV(20),XMJAT(20),XNUI(64),
23 COMMON BETA(64,20),ELJ(10,20),GELJ(10,20),
24 XNUIJ(64,20),XNUIJB(64,20),
25 COMMON IC,INEQ,INEQV,IP,IRUN,ISC,ISCP1,
26 ISMC,ISMCNR,ISR,ISS,ISSNR,ISSP1,ISSP2,ISSP3,ISSP4,
27 ISW1A,ISW1B,ISW2A,ISW2B,ISW3A,ISW3B,ISW4A,ISW4B,ISW5A,
28 ISW6A,ISW6B,IUPC,IZERO,IJK,LC,MI,NFIT,
29 NIT,NNN,NNS,NQS,NQT,NTEST
30 COMMON IGM(20),ITB(5),KUR(64,20),LPIJ(20,10)
31 COMMON ACCM,ELMENT,HP
32 DIMENSION AAA(22,24),BTA(64,20),CAPQ(31),CCI(20),DGJ(20),SBKES
33 J(20),SDGJ(20),SHJAP(20),THEVP(20)
34 EQUIVALENCE (AAA(1,1),AAA(1,1)),(BETA(1,1),BTA(1,1)),
35 (BLBK(1),CAPQ(1)),(CAPXTH(1),CCI(1)),(GJA(1),DGJ(1)),
36 (CAPX(1),GJ(1)),(SAJ(1),SHJ(1)),(SS(1),SDGJ(1)),
37 (SHJA(1),SHJAP(1)),(THEV(1),THEVP(1))
38 COMMON /RDMAIN/ HS,SUPGO,MFITER,NOTRAN
39 COMMON /MASSFL/ SMASS,CTMXXI,TSTOPI,IS(20)
40 COMMON /ERROR/ ERR
41 COMMON /ERROR/ ERR
42 DATA RNAME /6HRESTMP/
43 DIMENSION AM(3,3),BM(3),F(3),CTSAVE(3)
44 NAMELIST /RSTDMP/ N,CTSAVE,F,D1,D2,DCT,DMC,BM,CONST,SM,CMA,DCTAPO
45 NAMELIST /RSTD2P/ N,N2,N3,SMASS,HS,PRESA,CTAP,H,SMP,HO,CTAPP,
46 TLAST,PLAST
47 DATA TERR,PERR /2*0.0001/
48 DATA RGAS /8.31434E7/
49 IF (ISW5A.NE.0) WRITE (6,200)
50 N=0
51 IF (ISW2B.LT.0) GO TO 90
52 PCGS=PRESA*1.01325F6
53 CONST=RGAS*(SMASS/PCGS)**2
54 STANDARD GUESS FOR FIRST POINT
55 CTAP=10000.

```

```

10 N=N+1
   IF (N.GT.20) GO TO 160
   DO 20 I=1,2
     J=3-I
     CTSAVE(J+1)=CTSAVE(J)
     F(J+1)=F(J)
     CALL INIT
     CALL INTA
     IF (ERR) RETURN
     CALL NRVAX
     IF (ERR) RETURN
     CTSAVE(1)=CTAP
     F(1)=CMA*SM*SM
     IF (ISW5A.NE.0) WRITE (6,RSTOMP)
     IF (N-2) 30,40,50
     ITERATE TO GET SECOND POINT
     CTAP=EF(1)/CONST
     GO TO 10
   C LINEAR INTERPOLATION TO GET THIRD POINT
   20 DCT=CTSAVE(2)-CTSAVE(1)
     D1=(F(1)*CTSAVE(2)-F(2)*CTSAVE(1))/DCT
     D2=(F(2)-F(1))/DCT
     CTAP=-D1/(D2-CONST)
     DCTAPO=ABS(CTAP-CTSAVE(1))
     IF (DCTAPO/ABS(CTAP).LE.TERR) GO TO 190
     GO TO 10
   C QUADRATIC INTERPOLATION
   30 DO 60 I=1,3
     AM(I,1)=1.00000000
     AM(I,2)=CTSAVE(I)
     AM(I,3)=CTSAVE(I)*CTSAVE(I)
     BM(I)=F(I)
     CALL SIMQ (AM,BM,3,KS)
     IF (KS.NE.0) GO TO 40
     DVC=BM(2)-CONST
     IF (BM(3).NE.0.) GO TO 70
     CTAP=-BM(1)/DVC
     GO TO 80
   40 CTAP=(-DMC+SIGN(SORT((DMC*DMC-4.*BM(1)*BM(3)).DMC))*0.5/BM(3)
     DCTAP=ABS(CTAP-CTSAVE(1))
     IF (DCTAP/ABS(CTAP).LE.TERR) GO TO 190
     IF (DCTAP.GE.DCTAPO) GO TO 40
     DCTAPO=DCTAP
     GO TO 10
   C ENTHALPY-MASS FLOW OPTION
   50 PRESA=4.8E-3*SMASS*SORT(HS)
     HEQR=1.E-5*HS
     SMERR=1.E-5*SMASS
     CTAP=10000.
     N=N+1
     IF (N.GT.20) GO TO 160
     PLAST=CTAP
     PLAST=PRESA
     N2=0
     N2=N2+1

```

ORIGINAL PAGE IS
POOR QUALITY

```

RES 56
RES 57
RES 58
RES 59
RES 60
RES 61
RES 62
RES 63
RES 64
RES 65
RES 66
RES 67
RES 68
RES 69
RES 70
RES 71
RES 72
RES 73
RES 74
RES 75
RES 76
RES 77
RES 78
RES 79
RES 80
RES 81
RES 82
RES 83
RES 84
RES 85
RES 86
RES 87
RES 88
RES 89
RES 90
RES 91
RES 92
RES 93
RES 94
RES 95
RES 96
RES 97
RES 98
RES 99
RES 100
RES 101
RES 102
RES 103
RES 104
RES 105
RES 106
RES 107
RES 108
RES 109
RES 110

```

```

120 IF (N2.GT.20) GO TO 160
    HO=H
    CALL INIT
    CALL INTA
    IF (ERR) RETURN
    H=CHA*CRP/CMA
    IF (ISWSA.NE.O) WRITE (6,RSTDP2)
    IF (ABS(H-HS).LE.HERR) GO TO 130
    IF (N2.GT.1) GO TO 120
    CTAPP=CTAP
    CTAPE=CTAP*HS/H
    GO TO 110
    CTSV=CTAP
    CTAP=CTAP*(HS-H)*(CTAPP-CTAP)/(HO-H)
    CTAPP=CTSV
    GO TO 110
130 N3=0
140 N3=N3+1
    IF (N3.GT.20) GO TO 160
    CALL NRMX
    IF (ERR) RETURN
    SMP=SM*RHAP*SORT(RGAS*CTAP/CMA)
    IF (ISWSA.NE.O) WRITE (6,RSTDP2)
    IF (ABS(SMP-SMASS).LE.SMERR) GO TO 150
    PRESA=PRESA*SMASS/SMP
    CALL INTA
    IF (ERR) RETURN
    GO TO 140
150 IF (ABS(CTAP-TLAST).GT.TERR*TLAST) GO TO 100
    IF (ABS(PRESA-PLAST).GT.PERR*PLAST) GO TO 100
    GO TO 190
160 WRITE (6,210)
    IF (ISW2R.EQ.O) GO TO 170
    WRITE (6,RSTDP2)
    GO TO 180
170 WRITE (6,RSTDMP)
180 CALL DUMP (RNAME)
190 RETURN
200 C
210 C
2200 C
2310 C

```

-540-

```

TE=NT.EQ.1
LIM=ISSP1
IF (.NOT. TE) LIM=LIM+2
QDPER=CDPE
DQMRK=2.*DQMAX
IF (ISWSB.EQ.0) GO TO 10
ISTMT=1
WRITE (6,410) ISTMT
WRITE (6,RNKDMP)
DO 20 J=1,ISS
F1(J)=DGJ(J)
GJ1(J)=GJR(J)
GJ2(J)=GJ(J)
F1(ISSP1)=DT
GJ1(ISSP1)=CTB
CT=CTB+DT*HDELX
GJ2(ISSP1)=CT
IF (TE) GO TO 30
F1(ISSP2)=DTE
GJ1(ISSP2)=CTEB
CTE=CTEB+DTE*HDELX
GJ2(ISSP2)=CTE
F1(ISSP3)=DCHA
GJ1(ISSP3)=CHB
CHA=CHB+DCHA*HDELX
GJ2(ISSP3)=CHA
DO 40 J=1,LIM
IFAIL=-J
IF (GJ2(J)) 270,40,40
CONTINUE
CX=CXB+HDELX
CALL DERIVS
IF (ERR.OR.FAILED) RETURN
IF (ISWSB.EQ.0) GO TO 50
ISTMT=2
WRITE (6,410) ISTMT
WRITE (6,RNKDMP)
DO 60 J=1,ISS
F2(J)=DGJ(J)
GJ3(J)=GJ(J)
F2(ISSP1)=DT
CT=CTB+DT*HDELX
GJ3(ISSP1)=CT
IF (TE) GO TO 70
F2(ISSP2)=DTE
CTE=CTEB+DTE*HDELX
GJ3(ISSP2)=CTE
F2(ISSP3)=DCHA
CHA=CHB+DCHA*HDELX
GJ3(ISSP3)=CHA
DO 80 J=1,LIM
IFAIL=-J

```

10

20

30

40

50

60

70

ORIGINAL PAGE IS
OF POOR QUALITY

RNK 56
 RNK 57
 RNK 58
 RNK 59
 RNK 60
 RNK 61
 RNK 62
 RNK 63
 RNK 64
 RNK 65
 RNK 66
 RNK 67
 RNK 68
 RNK 69
 RNK 70
 RNK 71
 RNK 72
 RNK 73
 RNK 74
 RNK 75
 RNK 76
 RNK 77
 RNK 78
 RNK 79
 RNK 80
 RNK 81
 RNK 82
 RNK 83
 RNK 84
 RNK 85
 RNK 86
 RNK 87
 RNK 88
 RNK 89
 RNK 90
 RNK 91
 RNK 92
 RNK 93
 RNK 94
 RNK 95
 RNK 96
 RNK 97
 RNK 98
 RNK 99
 RNK 100
 RNK 101
 RNK 102
 RNK 103
 RNK 104
 RNK 105
 RNK 106
 RNK 107
 RNK 108
 RNK 109
 RNK 110

```

80  IF (GJ3(J)) 270.80.80
    CONTINUE
    IF (TE) GO TO 90
    IFAIL=ISSP4
    IF (ABS(QDPE-QCPEB).LE.DQMRK.OR.DOMAX.EQ.0.) GO TO 90
    GO TO 270
90  IFAIL=0
    CALL DERIVS
    IF (ERR.OR.FAILED) RETURN
    IF (ISWSB.EQ.0) GO TJ 100
    ISTMNT=3
    WRITE (6.410) ISTMNT
    WRITE (6.RNKDMP)
    DO 110 J=1,ISS
    100 F3(J)=DGJ(J)
    110 F3(ISSP1)=DT
    IF (TE) GO TO 120
    F3(ISSP2)=DTE
    F3(ISSP3)=DCHA
    DO 150 J=1,LIM
    120 IF (F1(J).EQ.0.) GO TO 130
    130 X4(J)=F2(J)-F1(J)
    IF (ABS(X4(J)/F1(J)).GE.0C01) GO TO 140
    P(J)=0.0
    GO TO 150
    140 XX=ABS(F3(J)-F2(J))
    IF (XX.EQ.0.) GO TO 130
    XX=ALOG10(XX)
    YY=ALOG10(DABS(X4,J))
    IF ((XX-YY).GT.38.0) GO TO 130
    P(J)=2.C*(F3(J)-F2(J))/X4(J)
    CONTINUE
    DO 210 J=1,LIM
    150 IF (P(J)) 160.210.210
    160 IF (P(J)+1.25) 200.200.170
    170 Z=3.0
    S1=0.0
    S2=0.0
    S3=0.0
    T1=1.0
    T2=0.5
    T3=1.0/6.0
    S1=S1+T1
    S2=S2+T2
    S3=S3+T3
    Z=Z+1.0
    T1=T1+P(J)/(Z-2.0)
    T2=T2+P(J)/(Z-1.0)
    T3=T3+P(J)/7
    IF ((S3+T3)-S3) 180.190.180
    180 X1(J)=S1
    X2(J)=S2
    X3(J)=S3
    GO TO 210
    190 X1(J)=(EXP(P(J))-1.0)/P(J)
    200

```

```

RNK 111
RNK 112
RNK 113
RNK 114
RNK 115
RNK 116
RNK 117
RNK 118
RNK 119
RNK 120
RNK 121
RNK 122
RNK 123
RNK 124
RNK 125
RNK 126
RNK 127
RNK 128
RNK 129
RNK 130
RNK 131
RNK 132
RNK 133
RNK 134
RNK 135
RNK 136
RNK 137
RNK 138
RNK 139
RNK 140
RNK 141
RNK 142
RNK 143
RNK 144
RNK 145
RNK 146
RNK 147
RNK 148
RNK 149
RNK 150
RNK 151
RNK 152
RNK 153
RNK 154
RNK 155
RNK 156
RNK 157
RNK 158
RNK 159
RNK 160
RNK 161
RNK 162
RNK 163
RNK 164
RNK 165

```

210	X2(J)=(X1(J)-1.0)/P(J)	RNK 166
	X3(J)=(X2(J)-0.5)/P(J)	RNK 167
	CONTINUE	RNK 168
220	DO 240 J=1,LIM	RNK 169
	IF (P(J)) 230,220,220	RNK 170
	G4(J)=GJ1(J)+F3(J)*DELTA	RNK 171
	IFOR=1	RNK 172
230	GO TO 240	RNK 173
	GJ4(J)=GJ1(J)+DELTA*(2.0*F3(J)*X2(J)+F1(J)*(X1(J)-2.0*X2(J))-F2(J)	RNK 174
	1)*F(J)*X2(J))	RNK 175
	IFOR=2	RNK 176
240	CONTINUE	RNK 177
	DO 250 J=1,LIM	RNK 178
	IFAIL=J-30*IFOR	RNK 179
250	IF (GJ4(J)) 270,250,250	RNK 180
	CONTINUE	RNK 181
	IF (TE) GO TO 260	RNK 182
	IFAIL=-ISSP4-30*IFOR	RNK 183
	IF (ABS(QDPE-QDPEB).LE.QDMPK.OR.QDMPK.EQ.0.) GO TO 260	RNK 184
	GO TO 270	RNK 185
260	IFAIL=0	RNK 186
	GO TO 280	RNK 187
270	FAILED=.TRUE.	RNK 188
	GO TO 400	RNK 189
280	DO 290 J=1,ISS	RNK 190
290	GJ(J)=GJ4(J)	RNK 191
	CT=GJ4(ISSP1)	RNK 192
	IF (TE) GO TO 300	RNK 193
	CTE=GJ4(ISSP2)	RNK 194
	CHA=GJ4(ISSP3)	RNK 195
	CX=CX8+DELTA	RNK 196
300	CALL DEPIVS	RNK 197
	IF (EAF.EQ.0) RETURN	RNK 198
	IF (ISSP1.EQ.0) GO TO 310	RNK 199
	ISTMT=1	RNK 200
	WRITE (410) ISTMT	RNK 201
	WRITE (6,RNKDMP)	RNK 202
310	DO 320 J=1,ISS	RNK 203
320	F4(J)=GJ(J)	RNK 204
	F4(ISSP1)=DT	RNK 205
	IF (TE) GO TO 330	RNK 206
	F4(ISSP2)=OTE	RNK 207
	F4(ISSP3)=DCHA	RNK 208
330	DO 370 J=1,LIM	RNK 209
	IF (P(J)) 350,340,340	RNK 210
340	SDQ(J)=(DELTA/6.0)*(F1(J)+2.0*F2(J)+2.0*F3(J)+F4(J))	RNK 211
	GO TO 370	RNK 212
	DO 360 K=1 4	RNK 213
350	Y(K)=F(K,J)-(P(J)*GJK(J,K))/DELTA	RNK 214
360	SDQ(J)=DELTA*(F1(J)*X1(J)+(-3.0*Y1+2.0*Y2+2.0*Y3-Y4)*X2(J)+4.0*(Y	RNK 215
	11-Y2-Y3+Y4)*X3(J))	RNK 216
370	CONTINUE	RNK 217
	DO 380 J=1,ISS	RNK 218
380	SDGJ(J)=SDQ(J)	RNK 219
	SDT=SDQ(ISSP1)	RNK 220

```

IF (TE) GO TO 390
SDTE=SDQ(ISSP2)
SOCHA=SDQ(ISSP3)
IF (ISW58.EQ.0) GO TO 400
ISTMNT=5
WRITE (6,410) ISTMNT
WRITE (6,RNKDMP)
WRITE (6,RNKDP2)
RETURN

```

390

400

C
C
C

410

```

FORMAT (18H DUMP AT STATEMENT,12)
END

```

```

RNK 221
RNK 222
RNK 223
RNK 224
RNK 225
RNK 226
RNK 227
RNK 228
RNK 229
RNK 230
RNK 231
RNK 232
RNK 233
RNK 234-

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

1 SUBROUTINE SHOCK (BMF,DEL,GAM,FL,IERR)
2 KEEP THIS IMPLICIT STATEMENT IN THE UN. AC VERSION
3 IMPLICIT REAL*8 (A-H,O-Z)
4 HARVEY BUSS ROUTINE - OBLIQUE SHOCK IN A PERFECT GAS
5
6 *****DEFINITIONS OF ARGUMENTS*****
7 BMF=FREE STREAM MACH NUMBER
8 DEL=WEDGE HALF ANGLE (DEGREES)
9 GAM=RATIO OF SPECIFIC HEATS
10 FL(1)=SHOCK ANGLE (RADIAN)
11 FL(2)=MACH NUMBER BEHIND SHOCK
12 FL(3)=DENSITY RATIO, RHO(BEHIND)/RHO(INFINITY)
13 FL(4)=STATIC PRESSURE RATIO, P(BEHIND)/P(INFINITY)
14 FL(5)=STATIC TEMPERATURE RATIO, T(BEHIND)/T(INFINITY)
15 FL(6)=TOTAL PRESSURE RATIO, PT(BEHIND)/PT(INFINITY)
16 FL(7)= TSIAG/TINF
17 IERR=1 NORMAL RETURN, IERR=2 ERROR RETURN, IERR=0 DETACHED SHOCK
18
19 DIMENSION FL(7)
20 KIT=0
21 DELR=DEL/57.295779
22 SD=SIN(DELR)
23 CD=COS(DELR)
24 BM2=BMF**2
25 DM4=BM2**2
26 C1=GAM+1.0
27 C2=GAM-1.0
28 B=-(BM2+2.0)/BM2-GAM*SD**2
29 C=(2.*4*2+1.)/BM4+(C1**2/4.+C2/RM2)*SD**2
30 DE=CD**2/BM4
31 C3=2.*B
32 UG=0.
33 FU=D+UG*(C+UG*(B+UG))
34 FRU=C+UG*(C3+3.*UG)
35 UN=UG-FU/FRU
36 IF (DABS(UG/UN-1.DC)-1.0-8) 50.50.20
37 IF (KIT-50) 30.30.40
38 KIT=KIT+1
39 UG=UN
40 GO TO 10
41 WRITE (6,100)
42 IERR=2
43 RETURN
44 ALP=B+UN
45 BETA=UN*ALP+C
46 DIS=ALP**2-4.*BETA
47 IF (DIS.GE.0.) GO TO 60
48 WRITE (6,110)
49 IERR=0
50 RETURN
51 DIS=SQRT(DIS)
52 U1=(-ALP+DIS)/2.0
53 U2=(-ALP-DIS)/2.0
54 IF (U1-U2) 70.70.80
55 RT=U1

```

```

80      GO TO 90
C      RT=U2
90      80 ZT= SORT(RT)
C      CONTINUE
C      FL(1)=ATAN(ZT/SORT(1.-ZT**2))
C      V1=BM2*RT
C      V2=C1**2*BM4*RT-4.0 *(V1-1.0 )*(GAM*V1+1.0 )
C      V3=(2.0 *GAM*V1-C2)*(C2*V1+2.0)
C      FL(2)= SORT(V2/V3)
C      FL(3)=C1*V1/(C2*V1+2.0)
C      FL(5)=V3/C1**2/V1
C      FL(4)=(2.0*GAM*V1-C2)/C1
C      U4=C2/2.0
C      BME2=FL(2)**2
C      CS=GAM/C2
C      SMFP=1.0 +C4*BM2
C      SMFP=1.0 +C4*BME2
C      FL(6)=(SMFP/SMFP)**C5*FL(4)
C      FL(7)=SMFP
C      IERR=1
C      RETURN
100     FORMAT (32H EXCEEDED 50 ITERATIONS IN SHOCK)
110     FORMAT (110H ----SHOCK DETACHED IN CLASSICAL WEDGE CALCULATION. UN
        1MODIFIED CHENG-KEMP THEORY IS USED FOR REMAINING ANGLES.)
        END

```

```

SHK 56
SHK 57
SHK 58
SHK 59
SHK 60
SHK 61
SHK 62
SHK 63
SHK 64
SHK 65
SHK 66
SHK 67
SHK 68
SHK 69
SHK 70
SHK 71
SHK 72
SHK 73
SHK 74
SHK 75
SHK 76
SHK 77
SHK 78
SHK 79
SHK 80
SHK 81-

```

ORIGINAL PAGE IS
OF POOR QUALITY

SPECIAL CASE N=1

IF (DABS(A(1))*.LE.TOL) GO TO 40
B(1)=B(1)/A(1)
RETURN

FORWARD SOLUTION

JJ=-N
DO 90 J=1,N
JY=J+1
JJ=JJ+N+1
BIGA=0
IT=JJ-J
DO 30 I=J,N

SEARCH FOR MAXIMUM COEFFICIENT IN COLJMN

IJ=IT+1
IF (DABS(BIGA)-DABS(A(IJ))) 20,30,30
BIGA=A(IJ)
IMAX=I
CONTINUE

TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)

IF (DABS(BIGA)-TOL) 40,40,50
KS=1
RETURN

INTERCHANGE ROWS IF NECESSARY

I1=J+N*(J-2)
IT=IMAX-J
DO 60 K=J,N
I1=I1+N
I2=I1+IT
SAVE=A(I1)
A(I1)=A(I2)
A(I2)=SAVE

DIVIDE EQUATION BY LEADING COEFFICIENT

A(I1)=A(I1)/BIGA
SAVE=B(IMAX)
B(IMAX)=B(J)
B(J)=SAVE/BIGA

ELIMINATE NEXT VARIABLE

IF (J-N) 70,100,70
IQS=N*(J-1)
DO 90 IX=JY,N
IXJ=IQS+IX
IT=J-IX

ORIGINAL PAGE IS
OF POOR QUALITY

SIM 56
SIM 57
SIM 58
SIM 59
SIM 60
SIM 61
SIM 62
SIM 63
SIM 64
SIM 65
SIM 66
SIM 67
SIM 68
SIM 69
SIM 70
SIM 71
SIM 72
SIM 73
SIM 74
SIM 75
SIM 76
SIM 77
SIM 78
SIM 79
SIM 80
SIM 81
SIM 82
SIM 83
SIM 84
SIM 85
SIM 86
SIM 87
SIM 88
SIM 89
SIM 90
SIM 91
SIM 92
SIM 93
SIM 94
SIM 95
SIM 96
SIM 97
SIM 98
SIM 99
SIM 100
SIM 101
SIM 102
SIM 103
SIM 104
SIM 105
SIM 106
SIM 107
SIM 108
SIM 109
SIM 110

```

DO 80 JX=JY,N
IXJX=N*(JX-1)+IX
JJX=IXJX+IT
A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
B(IX)=B(IX)-(B(J)*A(IXJ))
      BACK SOLUTION
NY=N-1
IT=N*N
DO 110 J=1,NY
IA=IT-J
IB=N-J
IC=N
DO 110 K=1,J
B(IB)=B(IB)-A(IA)*B(IC)
IA=IA-N
IC=IC-1
RETURN
END

```

80
90
C
C
C
100

110

111
SIM 112
SIM 113
SIM 114
SIM 115
SIM 116
SIM 117
SIM 118
SIM 119
SIM 120
SIM 121
SIM 122
SIM 123
SIM 124
SIM 125
SIM 126
SIM 127
SIM 128
SIM 129
SIM 130-


```

LIM1=2
CTMX=-1000.
DO 40 K=1.4
DO 30 I=1.74
TPROP(I,K,1)=0.
CONTINUE
DO 50 M=LIM1,LIM
  TV=0
  II=1
  TV=0.
  DO 70 I=1.74
    IT=IT+1
    IZ=I
    IF (IT.LE.NDT(II)) GO TO 60
    II=II+1
    IT=I
    TV=TV+DELTT(II)
    IF (M.EQ.1) TT(I)=TV
    CT=TV/CTAP
    CALL THERM
    TPROP(I,1,M)=XMJAT(J)-SHJA(J)/CT
    TPROP(I,2,M)=(SHJ(J)-SHJA(J))*CRP/1000.
    TP P(I,3,M)=CCPJ(J)*CRA
    I DP(I,4,M)=SFNT(J)*CRA
    IF (CT-1.) 70.9C.80
    CONTINUE
  CTMX=-1000.
  DO 100 I=1.1Z
    WRITE (6,320) TT(I),((TPROP(I,K,M),K=1.4),M=1.LIM)
    CONTINUE
  RETURN
ENTRY STUNT2
WRITE (6,330)
WRITE (6,340)
WRITE (6,350)
DO 150 L=1,NKQ
  I=ISEQ(L)
  NNQI=NNQ(I)
  DO 140 J=1,NNQI
    IIQJ=IIQ(J,I)
    JJQJ=JJQ(J,I)
    IF (J-1) 120,120,130
    WRITE (6,360) L,I,KKQ(I),(VV(K,I),K=1.5),SPRP(1,IIQJ),SPRP(1,JJQJ)
    GO TO 140
    WRITE (6,370) SPRP(1,IIQJ),SPRP(1,JJQJ)
    CONTINUE
  WRITE (6,330)
  WRITE (6,390)
  WRITE (6,350)
  MV2=1
  LQ2=0
  DO 190 L=1,NKQ
    KK=KKQ(L)

```

```

56 STU
57 STU
58 STU
59 STU
60 STU
61 STU
62 STU
63 STU
64 STU
65 STU
66 STU
67 STU
68 STU
69 STU
70 STU
71 STU
72 STU
73 STU
74 STU
75 STU
76 STU
77 STU
78 STU
79 STU
80 STU
81 STU
82 STU
83 STU
84 STU
85 STU
86 STU
87 STU
88 STU
89 STU
90 STU
91 STU
92 STU
93 STU
94 STU
95 STU
96 STU
97 STU
98 STU
99 STU
100 STU
101 STU
102 STU
103 STU
104 STU
105 STU
106 STU
107 STU
108 STU
109 STU
110 STU

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

160 MV1=MV2+1
    MV2=MV2+NV(KK)
    LQ1=LQ2+1
    LQ2=NO(L)
    WRITE (6,390) L,KK,(V(I),I=MV1,MV2)
    IF (LQ2.LI.LQ1) GO TO 190
    DO 190 LQ=LQ1,LQ2
    I=IQ(LQ)
    J=JQ(LQ)
    IF (LQ-LQ1) 160,160,170
    WRITE (6,400) HP(I),HP(J)
    GO TO 190
170 WRITE (6,370) HP(I),HP(J)
180 CONTINUE
190 CONTINUE
    IF (ISWIB.GT.0) GO TO 240
    WRITE (6,250)
    ICARD=0
    DO 200 I=1,ISS
    SAVEC(I)=GJA(I)*CMA
    DO 230 I=1,ISS
    DO 220 J=1,ISS
    WRITE (6,260)
    DO 210 K=1,30
    TV=1000*K
    IF (TV.GT.1.00001*CTAP) GO TO 220
    ICARD=ICARD+1
    CALL TRANSP (TV,PRESA)
    WRITE (6,270) TV,(G(L,I,J).L=1,3),I,J,HP(I),HP(J),ICARD
    IF (ISW19.NE.-1) GO TO 210
    PUNCH 270, TV,(G(L,I,J).L=1,3),I,J,HP(I),HP(J),ICARD
    CONTINUE
    CONTINUE
    CONTINUE
    WRITE (6,280)
    RETURN

210
220
230
240
    C
    C
    C
250
    FORMAT (1H1,38X,56H AVERAGED PAIR CROSS-SECTIONS AS FUNCTIONS OF TEMPERATURE)
    FORMAT (//2X,5HTEMP.,8X,4HQ(1),8X,4HQ(2),8X,4HQ(3),5X,7HINDICES,5STU 151
    1X,7HSPECIES,8X,5HCOUNT)
    FORMAT (F7.0,1P3E12,3,5X,13,1H-,13,3X,/4,1H-,A4,5X,18)
    FORMAT (//10H NOTE - CHARGED SPECIES CROSS SECTIONS AT ALL TEMPERATURES ARE BASED ON ELECTRON MOLE FRACTION IN RESERVOIR.)
    FORMAT (1H1,40X,22HTHERMAL PROPERTIES OF ,A4,6H (H00=.F10,3,11H KCSTU 157
    1AL/MOLE))
    FORMAT (1H1,10X,3A6,18H**PHYSICAL MODEL**,3A6/1H0,8X,1HT,2X,12H (MUSTU 159
    10-H00)/RT,9X,5HH-H00,12X,2HCP,12X,2HS0/29X,9HKCAL/MOLE,2X,12HCAL/MSTU 160
    20LE-DEG,2X,12HCAL/MOLE-DEG)
    FORMAT (1H0,11X,3A6,18H**PHYSICAL MODEL**,3A6,12X,3A6,18H**THERMSTU 161
    10 FIT**,3A6/1H0,8X,1HT,2X,12H(MUQ-H00)/RT,9X,5HH-H00,12X,2HCP,12STU 162
    2X,2HS0,12X,12H(MUQ-H00)/RT,9X,5HH-H00,12X,2HCP,12X,2HS0/29X,9HKCALSTU 163
    3/MOLE,2X,12HCAL/MOLF-DEG,2X,12HCAL/MOLF-DEG,29X,9HKCAL/MOLE,2X,12HSTU 165

```


320	4CAL/MOLE-DEG.2X.12HCAL/MOLE-DEG)	STU 166
330	FORMAT (F10.0,4F14.3,1CX,4F14.3)	STU 167
340	FORMAT (1H1.4PX.28HTRANSPORT CROSS SECTION DATA)	STU 168
350	FORMAT (60X.7H(INPUT))	STU 169
	FORMAT (1H0.5X.4HSTEP.5X.SHINDEX.4X.6HOPTION.1CX.SHVV(1),1CX.SHVV(STU 170
360	12),1CX.SHVV(3),1CX.SHVV(4),1CX.SHVV(5),9X.1HINTERACTION)	STU 171
370	FORMAT (2110.1P5E15.3,9X.A4.3H - .A4)	STU 172
380	FORMAT (114X.A4.3H - .A4)	STU 173
390	FORMAT (59X.8H(EDITED))	STU 174
400	FORMAT (110.120.1P5E15.3)	STU 175
	FORMAT (1.14.113X.A4.3H - .A4)	STU 176
	END	STU 177-


```

30      MIX=.TRUE.
      WTF=C.001*(ZT+500.)
      WTHO=1.-WTF
      GO TO 50
40      MIX=.FALSE.
      IF (ETAJ(J)) 50.50.70
      MIX=.FALSE.
50      IF (CT-CTNXX) 70.50.50
      IF (E1) GO TO 60
      C
      C
      THERMO FIT
      XMJAT(J)=TFA(J)*(1.-CLNT)-TFK(J)-CT*(TFB(J)+CT*(TFC(J)/2.+CT*(TFD(
1J)/3.+CT*TFE(J)/4.)))+SHJA(J)/CT
      SHJ(J)=SHJA(J)+CT*(TFA(J)+CT*(TFB(J)+CT*(TFC(J)+CT*TFE(
1J))))
      CCPJ(J)=TFA(J)+CT*(2.0*TFB(J)+CT*(3.0*TFC(J)+CT*(4.0*TFD(J)+CT*(5.
1J*TFE(J))))
      IF (1.NOT.MIX) GO TO 300
      XMTF=XMJAT(J)
      SHTF=SHJ(J)
      CCPTF=CCPJ(J)
      STF=STF/CT-XMTF
      C
      C
      C
      PHYSICAL MODEL
      CCPJ(J)=0.
      IF (E1) GO TO 80
      SHJ(J)=0.
      XMJAT(J)=0.
      IF (ETAJ(J).LT.3.) GO TO 150
      TRANSJATIONAL, ROTATIONAL, AND VIBRATIONAL CONTRIBUTIONS FOR
      LINEAR TRIATOMIC SPECIES
      Z1=3.5*CLNT
      DO 140 K=1,4
      IF (INEQV) 90.100.90
      Z3=THEVE(K,J)
      GO TO 110
      Z3=THEVE(K,J)/CT
      IF (Z3.GT.88.) GO TO 140
      Z4=EXP(Z3)
      Z5=Z4-1.
      Z6=THEVE(K,J)/Z5
      IF (INEQV.NE.1ZER0) GO TO 120
      Z7=Z3/Z5
      GO TO 130
      Z7=0.
      Z2=SZ2+Z6/CT
      Z8=ALOG(Z4/Z5)
      Z9=Z7*(Z4*Z7)
      CCPJ(J)=CCPJ(J)+Z9
      IF (E1) GO TO 140
      XMJAT(J)=XMJAT(J)-Z8
      SHJ(J)=SHJ(J)+Z6
      CONTINUE
140

```

```

THE 56
THE 57
THE 58
THE 59
THE 60
THE 61
THE 62
THE 63
THE 64
THE 65
THE 66
THE 67
THE 68
THE 69
THE 70
THE 71
THE 72
THE 73
THE 74
THE 75
THE 76
THE 77
THE 78
THE 79
THE 80
THE 81
THE 82
THE 83
THE 84
THE 85
THE 86
THE 87
THE 88
THE 89
THE 90
THE 91
THE 92
THE 93
THE 94
THE 95
THE 96
THE 97
THE 98
THE 99
THE 100
THE 101
THE 102
THE 103
THE 104
THE 105
THE 106
THE 107
THE 108
THE 109
THE 110

```

111 THE 112
 112 THE 113
 113 THE 114
 114 THE 115
 115 THE 116
 116 THE 117
 117 THE 118
 118 THE 119
 119 THE 120
 120 THE 121
 121 THE 122
 122 THE 123
 123 THE 124
 124 THE 125
 125 THE 126
 126 THE 127
 127 THE 128
 128 THE 129
 129 THE 130
 130 THE 131
 131 THE 132
 132 THE 133
 133 THE 134
 134 THE 135
 135 THE 136
 136 THE 137
 137 THE 138
 138 THE 139
 139 THE 140
 140 THE 141
 141 THE 142
 142 THE 143
 143 THE 144
 144 THE 145
 145 THE 146
 146 THE 147
 147 THE 148
 148 THE 149
 149 THE 150
 150 THE 151
 151 THE 152
 152 THE 153
 153 THE 154
 154 THE 155
 155 THE 156
 156 THE 157
 157 THE 158
 158 THE 159
 159 THE 160
 160 THE 161
 161 THE 162
 162 THE 163
 163 THE 164
 164 THE 165

ORIGINAL PAGE IS
 OF POOR QUALITY

```

CCPJ(J)=CCPJ(J)+3.5
IF (E1) GO TO 240
SZ1=SHJ(J)
XMJAT(J)=XMJAT(J)-SAJ(J)-Z1+SHJA(J)/CT+(1.5+ETAJ(J))*CTPL
SHJ(J)=SHJ(J)+3.5*CT+SHJA(J)
GO TO 240
TRANSLATIONAL, ROTATIONAL, AND VIBRATIONAL CONTRIBUTIONS FOR
MONATOMIC AND DIATOMIC SPECIES
Z1=ETAJ(J)-1.0
Z2=Z1+2.5
IF (E1) GO TO 160
XMJAT(J)=0.
CCPJ(J)=0.
IF (INEQV) 170,180,170
Z3=THEV(J)
GO TO 190
Z3=THEV(J)/CT
IF (Z3.GT.88.) GO TO 230
Z4=EXP(Z3)
Z5=Z4-1.
IF (Z1) 200,230,200
Z6=THEV(J)/Z5
IF (INEQV.NE.IZERO) GO TO 210
Z7=Z3/Z5
GO TO 220
Z7=C.0
Z8=ALOG(Z4/Z5)
Z9=Z7*(Z4*Z7)
CCPJ(J)=Z1*Z9
IF (E1) GO TO 230
XMJAT(J)=-(Z1*Z8)
CCPJ(J)=CCPJ(J)+Z2
IF (E1) GO TO 240
SHJ(J)=Z2*CT+SHJA(J)+Z1*Z6
XMJAT(J)=XMJAT(J)+SHJA(J)/CT-(SAJ(J)+Z2*CLGT)
IF (IGM(J).EQ.0) GO TO 290
ELECTRONIC CONTRIBUTIONS (ALL SPECIES)
S1=C.0
S2=C.0
S3=C.0
N=1*(J)
DO 250 M=1,N
XX=GELJ(M,J)*EXP(-ELJ(M,J)/CT)
S1=S1+XX
XX=X*ELJ(M,J)
S2=S2+XX
XX=X*ELJ(M,J)
S3=S3+XX
IF (E1) GO TO 260
XMJAT(J)=XMJAT(J)-ALOG(S1)
SHJ(J)=SHJ(J)+S2/S1
IF (S2.GT.1.0-37) GO TO 270
XX=C.0
GO TO 280
XX=S2*S2
  
```

```

280 CCBJ(J)=CCPJ(J)+(S1*S3-WW)/((S1*CT)**2)
290 IF (INEOV.FO.J) GO TO 300
    IF (ETAJ(J).GT.2.1) GO TO 300
    IF (E1) GO TO 310
    C      NTHOBY CALCULATION
    SENT(J)=22+(SHJA(J)+S2/S1)/CT+Z1*Z6-XMJAT(J)
    GO TO 310
    IF (E1) GO TO 310
    SENT(J)=SHJ(J)/CT-XMJAT(J)
    IF (INEOV.N.O) SENT(J)=SENT(J)+SZ1-SZ2
    IF (.NOT.MIX) GO TO 320
    C      MIX PHYSICAL MODEL AND THERMO FIT RESULTS IN TRANSITION
    C      TEMPERATURE RANGE
    CCBJ(J)=WTHO*CCPJ(J)+WTF*CCPTF
    IF (E1) GO TO 320
    XMJAT(J)=WTHO*XMJAT(J)+WTF*XMJTF
    SHJ(J)=WTHO*SHJ(J)+WTF*SHJTF
    SENT(J)=WTHO*SENT(J)+WTF*STF
    C      CONTINUE
    RETURN
    END
310
320

```

```

THE 166
THE 167
THE 168
THE 169
THE 170
THE 171
THE 172
THE 173
THE 174
THE 175
THE 176
THE 177
THE 178
THE 179
THE 180
THE 181
THE 182
THE 183
THE 184
THE 185
THE 186-

```

```

SUBROUTINE THROAT
REAL ACOM(30),ELEMENT(10),HP(20)
LOGICAL ERR
DOUBLE PRECISION AA,CAPX,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON C, CARB, CSTA, CHA, CLNT, AR, ARBA, ARBB, BZERO,
1 CCR, CCRS, CTT, CT, CTAP, CMA, CRA, CTMX, CTMP,
2 CTO, CTPL, CTT, CX, CXH, CXMAX, DATEST, DBTEST, DELT1,
3 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
4 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
5 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
6 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
7 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
8 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
9 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
10 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
11 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
12 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
13 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
14 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
15 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
16 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
17 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
18 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
19 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
20 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
21 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
22 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
23 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
24 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
25 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
26 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
27 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
28 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
29 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
30 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
31 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
32 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
33 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
34 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
35 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
36 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
37 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
38 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
39 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
40 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
41 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
42 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
43 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
44 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
45 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
46 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
47 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
48 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
49 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
50 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
51 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
52 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
53 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
54 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,
55 DELT2, DELTAX, DLOGA, DLOGR, DLOGR, DLOGR, DLOGR, DLOGR,

```

10
20
C
C
C
30

2.4HRSA=.E10.3.4X.6MDELEL=.E10.3.5X.E10.3//)
END

THR 56
THR 57-

ORIGINAL PAGE IS
OF POOR QUALITY

[illegible]


```

20  DO 20 J=1,N
C      Q(1,I,J)=ZM1(I)+ZM1(J)
C      C(J,I)=(ZM1(I)-ZM1(J))/Q(1,I,J)
C      C(I,J)=SQRT(ZM1(I)*ZM1(J)/Q(1,I,J))
C      RETURN
C      ENTRY TRANSP CALCULATES TRANSPORT PROPERTIES FOR TEMPERATURE T,
C      PRESSURE P, AND MOLE FRACTIONS X(J),
C      NOTE - THE MOLE FRACTIONS ARE NOT NECESSARILY IN EQUILIBRIUM AT
C      THE TEMPERATURE T.
30  T=TTAB
C      SAVE CT AND CCP(J) FOR LATER USE.
C      CTSAVE=CT
C      CT=T/CTAP
C      DO 40 J=1,N
C      CSAVE(J)=CCPJ(J)
C      CALCULATE SPECIES SPECIFIC HEATS AT TEMPERATURE T.
C      CALL THERM1
C      CPTOT=0.
C      CMSAVE=0.
C      DO 50 J=1,N
C      X(J)=SAVEC(J)
C      DH(J)=CCPJ(J)-2.5
C      CPTOT=CCPJ(J)*X(J)+CPTOT
C      CMSAVE=CGI(J)*X(J)+CMSAVE
C      CCPJ(J)=CSAVE(J)
C      CT=CTSAVE
C      COMPUTE COLLISION CROSS SECTIONS. STORE IN Q ARRAY.
C      CALL PUTGIN (X(1)*P)
C      COMPUTE MATRIX ELEMENTS FOR TRANSPORT CALCULATIONS. STORE IN Q
C      ARRAY.
C      IF (IS*18,LT,0) RETURN
C      DO 60 I=1,N
C      DO 60 J=1,N
C      Q(1,I,J)=C(1,J)*Q(1,I,J)
C      Q(2,I,J)=C(1,J)*Q(2,I,J)
C      S=SQRT(T)
C      SSIG=1.3451E5/S
C      FLEWIS=CPTOT/Q(1,1,1,1)
C      IF (N.EQ.1) GO TO 80
C      DO 70 J=2,N
C      Q(3,J,J)=Q(2,J,J)
C      J1=J-1
C      DO 70 I=1,J1
C      Q(3,I,J)=3.4375*Q(1,I,J)-.75*C(1,J)*Q(3,I,J)
C      Q(1,J,I)=(1.-C(J,I)*.2)*(Q(3,I,J)-Q(2,I,J))
C      Q(2,J,I)=C(J,I)*(3.75*Q(1,I,J)-Q(3,I,J))
C      Q(3,J,I)=Q(3,I,J)-Q(2,J,I)-Q(1,J,I)
C      Q(3,I,J)=Q(3,I,J)+Q(2,J,I)-Q(1,J,I)
C      Q(2,J,I)=(1.6666667*Q(1,I,J)-Q(2,I,J))/(ZM2(I)+ZM2(J))
C      Q(2,I,J)=Q(2,I,J)
C      COMPUTE TRANSPORT PROPERTIES. VISC=VISCOSITY, PRF=FROZEN PRANDTL
C      NUMBER, SIGMA=ELECTRICAL CONDUCTIVITY, FLEWIS=ATOM-MOLECULE LEWIS
C      NUMBER.
C      DO 90 I=.N
C      R(1,I)=X(I)*Q(1,I,I)

```

TRA 56
 TRA 57
 TRA 58
 TRA 59
 TRA 60
 TRA 61
 TRA 62
 TRA 63
 TRA 64
 TRA 65
 TRA 66
 TRA 67
 TRA 68
 TRA 69
 TRA 70
 TRA 71
 TRA 72
 TRA 73
 TRA 74
 TRA 75
 TRA 76
 TRA 77
 TRA 78
 TRA 79
 TRA 80
 TRA 81
 TRA 82
 TRA 83
 TRA 84
 TRA 85
 TRA 86
 TRA 87
 TRA 88
 TRA 89
 TRA 90
 TRA 91
 TRA 92
 TRA 93
 TRA 94
 TRA 95
 TRA 96
 TRA 97
 TRA 98
 TRA 99
 TRA 100
 TRA 101
 TRA 102
 TRA 103
 TRA 104
 TRA 105
 TRA 106
 TRA 107
 TRA 108
 TRA 109
 TRA 110

TRA 111
 TRA 112
 TRA 113
 TRA 114
 TRA 115
 TRA 116
 TRA 117
 TRA 118
 TRA 119
 TRA 120
 TRA 121
 TRA 122
 TRA 123
 TRA 124
 TRA 125
 TRA 126
 TRA 127
 TRA 128
 TRA 129
 TRA 130-

```

B(1,2)=X(1)*Q(2,1,1)
CALL ELCOND (SIGMA,SSIG)
CALL KINT (ZKINT)
CALL KANDMU (ZK)
ZKINT=ZK(1)+ZKINT
VISC=S*ZK(2)*1.C-3
FLEWIS=FLEWIS/ZKINT*IAMBIP
PRF=8.3167*CPTOT*ZK(2)/ZKINT/CMSAVE
IF (ISW8B.EQ.0) RETURN
PRINT TRANSPORT PROPERTY VALUES ONLY IF ISW8B.NE.0.
WRITE (6,100)
WRITE (6,110) T,P,VISC,PRF,SIGMA,FLEWIS,CPTOT,CMSAVE
WRITE (6,110) (X(J),J=1,N)
RETURN

C
C
C
100 FORMAT (1H )
110 FORMAT (1P12E11.3)
END
  
```

ORIGINAL PAGE IS
 OF POOR QUALITY

```

SURROUTINE WEDGE
DOUBLE PRECISION ZETA,CAPGAM,ZZP,ZZPP,Z,AMD,AND,GAMD,FL
LOGICAL WEDGEM,AXISYM,FINAL
COMMON /RDXEKG/ ANGLE(10),RADLE(5),WX1,DWX,WXI(20),TWEDGE,WK,
      /WCD/ TSF
COMMON /BL/ DELEL(2),BLINT(2),XZERO,TWALL,CPWALL,VISROT,DIAM(2),
      /RQJDM,IPCINT
COMMON /TRPRDP/ VISC,PR,SIGMA,FLEWIS
COMMON /OUTPDP/ FVOUT(35),GJMF(20)
EQUIVALENCE (GAMMA,FVOUT(10)), (AM,FVOUT(8)), (VMUINF,FVOUT(14)),
      (TINF,FVOUT(2)), (RHOINF,FVOUT(5)), (UINF,FVOUT(7)),
      (PINF,FVOUT(A)), (REPF,FVOUT(12)), (GMW,FVOUT(13))
1 DIMENSION XW(10),PAW(10),QW(10),DELSTW(10),YS(10),ZTA(10),UN(2)
2 DIMENSION FL(7), APR(10), AST(2), ASTQ(10),CH(10)
DIVENSION GFAC(2),GFAK(2)
DATA UN / .,UN.,.UN.,.RNAME/5HWEDGE/, ZERO /0./, ANMIN /1.E-4/
DATA PI /3.1415927/, AST / .,.,.,.ALMIN /0.01/
DATA AE /1./, RGC /8.314E7/, SQTPI /1.7724539/
ITH=1
IAMAX=100
ISD=IABS(ISW9B)
H3=FVOUT(3)/1.P+0.5*(FVOUT(7)*30.48)**2/4.186E7
HW=CPWALL*TWEDGE
H3 AND HW ARE IN CAL/GM
H3RATIO=HW/H3
TRATIO=TWEDGE/TINF
SEAM=SQRT(0.5*GAMMA)
RU2=RMUINF*UINF**2
TREF=TSE*(1.+3.*HRATIO)/6.
CALL TRANSF (TREF,PINF)
CEVISC/VMUINF*TINF/TREF*0.067197
THE NUMERICAL FACTOR CONVERTS VISC FROM GM/CM-SEC TO LBM/FT-SEC
TAU=0.664+1.73*HRATIO
AE=(GAMMA+1.)/2.
A2=AE*A
A4=AP*A2
A5=AP*A
A7=AP*A5
G2=GAMMA*GAMMA
G3=GAMMA*A*G2
WK2=AK*WK
WK3=WK2*WK
WK4=WK2*WK
EPS=(GAMMA-1.)/(GAMMA+1.)
EPS2=EPS*EPS
OFAC1=0.03986*PINF*SQRT(RGC*TINF/(2.*PI*GMW))*AE
OFAC2=S**2*(GAMMA-0.5*(GAMMA+1.)*TRATIO)/(GAMMA-1.)
/FAC=16.*A7/G3*EPS2/WK4
GFC=WK/(2.*A2*EPS)
AV2=AK*A4
OFAC=AE/GAMMA*A4.*EPS2*AM2/WK2
CFAC=G2/A5*WK3/R./EPS
CHOFAC=0.332*A*EPS

```

1 WED
 2 WED
 3 WED
 4 WED
 5 WED
 6 WED
 7 WED
 8 WED
 9 WED
 10 WED
 11 WED
 12 WED
 13 WED
 14 WED
 15 WED
 16 WED
 17 WED
 18 WED
 19 WED
 20 WED
 21 WED
 22 WED
 23 WED
 24 WED
 25 WED
 26 WED
 27 WED
 28 WED
 29 WED
 30 WED
 31 WED
 32 WED
 33 WED
 34 WED
 35 WED
 36 WED
 37 WED
 38 WED
 39 WED
 40 WED
 41 WED
 42 WED
 43 WED
 44 WED
 45 WED
 46 WED
 47 WED
 48 WED
 49 WED
 50 WED
 51 WED
 52 WED
 53 WED
 54 WED
 55 WED

ORIGINAL PAGE IS
OF POOR QUALITY

```

Y5FAC=0.125*G2*WK3/A4/EP5
X5=533.*C*AM2/REPF
X5=12.*C*(EPS*TAU*AM*AM2)**2/REPF
WRITE (6,260) UN(ITH)
*WRITE (6,270) XM
IF (ITH,EO,2) WRITE (6,280) XS
ITHS=ITH
GFAC(1)=GFAC/AM
GFAC(2)=GAMMA*GFAC
DO 240 I=1,NRADLE
TN=2.*RADLE(1R)
IF (TN,EO,0.) TN=1.E-6
EN=AV*SQRT(C/(REPF*TN/12.))
TAUN=TAUN*EN
TAUN2=TAUN*TAUN
TAUN3=TAUN*TAUN2
TAUN4=TAUN*TAUN3
CHD=CHD*GFAC*TAUN*EN*TN
CTAK=CTAK*GFAC*TN/TAUN4
ZFAK=ZFAK*TAUN6/TN
Y5FAK=Y5FAK*TN/TAUN4
GFAC(1)=GFAC(1)/TAUN2
GFAC(2)=GFAC(2)/TAUN2
PEAK=PEAK*TAUN4
OMEGA=1./PEAK
DO 230 IA=1,NANGLE
IF (IR,GT,1.OR,ITH,EO,2) GO TO 30
ANDEAM
ANDEAMAX1(ANGLE( ANMIN)
GAMDEGAMMA
CALL SHOCK (AMD,AND,GAMD,FL,IERR)
IF (IERR,NE,0) GO TO 20
IAX=IA
GO TO 30
APR(IA)=FL(4)
IF (IFGR,EO,1) GO TO 30
CALL DUMP (RNAME)
RETURN
IF (IA,GT,IAMAX) ITH=2
FINAL=FALSE
ALPHA=ANGLE(IA)/57.2958
IFMFO
IF (ITH,EO,1) GO TO 40
CAPGAM=GFAC(2)*ALPHA
GO TO 50
CAPGAM=GFAC(1)*SOR*(GAMMA/A*AMAX1(APR(IA)-1.,ZERO))
IF (IP,GT,1) GO TO 50
SSAS=SSIN(AMAX1(ALPHA,ALMIN))
SSASQ=SSA**2
EXPV=EXP(-SSASQ)
EXPV=1.+EXP(SSA)
QEM(IA)=QEMCI*(QFMC2*(EXPV+SOTPI*SSA*ERFV)-0.5*EXPV)
IT=0
PAI=1

```

WED 56
WED 57
WED 58
WED 59
WED 60
WED 61
WED 62
WED 63
WED 64
WED 65
WED 66
WED 67
WED 68
WED 69
WED 70
WED 71
WED 72
WED 73
WED 74
WED 75
WED 76
WED 77
WED 78
WED 79
WED 80
WED 81
WED 82
WED 83
WED 84
WED 85
WED 86
WED 87
WED 88
WED 89
WED 90
WED 91
WED 92
WED 93
WED 94
WED 95
WED 96
WED 97
WED 98
WED 99
WED 100
WED 101
WED 102
WED 103
WED 104
WED 105
WED 106
WED 107
WED 108
WED 109
WED 110

```

100 NW2=1
110 ITAB=1
120 IT=IT+1
130 IF (NW1.GT.NWX) GO TO 70
140 WXV1=WX1+(NW1-1)*DWX
150 GO TO 80
160 WXV1=1.E30
170 IF (NW2.GT.20) GO TO 90
180 WXV2=WX1(NW2)
190 GO TO 100
200 WXV2=1.E30
210 IF (WXV1-WXV2) 110,110,120
220 IF (WXV1.GT.1.E19) GO TO 130
230 XW(IT)=WXV1
240 NW1=NW1+1
250 GO TO 140
260 IF (WXV2.GT.1.E19) GO TO 130
270 XW(IT)=WXV2
280 NW2=NW2+1
290 GO TO 140
300 IF (IT.LE.1) GO TO 230
310 IT=IT-1
320 FINAL=.TRUE.
330 GO TO 120
340 XC=XW(IT)*COS(ALPHA)
350 ZETA=ZTA*XC
360 ZTA(IT)=ZETA
370 CALL RESOLN (ZETA,CAPGAM,ZZP,ZZPP,Z)
380 ZZPS=ZP
390 IF (ITH.EQ.1) GO TO 150
400 (CHENG-KEMP THEORY
410 YS(IT)=YSEAK*Z
420 PW(IT)=DINF*PFAK*ZZPP
430 SQT=SQRT(ZZPS)
440 DELSTW(IT)=DFAK*SQT/ZZPP
450 GO TO 150
460 MODIFIED CHENG-KEMP THEORY
470 SQT=SQRT(ZZPS+CMEGA*7IA(IT))
480 DELSTW(IT)=DFAK*SQT/(ZZPP+OMEGA)
490 PRATIO=PFAK*ZZPP+1.
500 PW(IT)=PINF*PRATIO
510 YS(IT)=YSEAK*(1.+SQT)/(ZZPP+OMEGA)+A*XC*TAN(ALPHA)
520 CH(IT)=CHD/DELSTW(IT)
530 QW(IT)=CH(IT)*PHOINF*UINF*(H0-HW)*1.8
540 ASTQ(IT)=AST(1)
550 IF (ITH.EQ.2) GO TO 170
560 IF (QAW(IT).L1.CFM(IA)) GO TO 170
570 IFM=1
580 QW(IT)=QFM(IA)
590 ASTQ(IT)=AST(2)
600 IF (IT.LI.10) GO TO 60
610 IF (ITAB.EQ.1) GO TO 190
620 WRITE (6,290)
630 GO TO 200
640 CPGM=CAPGAM

```

```

WED 111
WED 112
WED 113
WED 114
WED 115
WED 116
WED 117
WED 118
WED 119
WED 120
WED 121
WED 122
WED 123
WED 124
WED 125
WED 126
WED 127
WED 128
WED 129
WED 130
WED 131
WED 132
WED 133
WED 134
WED 135
WED 136
WED 137
WED 138
WED 139
WED 140
WED 141
WED 142
WED 143
WED 144
WED 145
WED 146
WED 147
WED 148
WED 149
WED 150
WED 151
WED 152
WED 153
WED 154
WED 155
WED 156
WED 157
WED 158
WED 159
WED 160
WED 161
WED 162
WED 163
WED 164
WED 165

```

```

200 WRITE (6,300) RADLE(IR),ANGLE(IA),CPGM, FGA
    WRITE (6,310) (XW(J),J=1,IT)
    WRITE (6,320) (PW(J),J=1,IT)
    WRITE (6,330) (CWW(J),ASTO(J),J=1,IT)
    WRITE (6,340) (CH(J),J=1,IT)
    WRITE (6,350) (DELSTW(J),J=1,IT)
    IF (IS.EQ.0) GO TO 220
    IF (IS.EQ.2) GO TO 210
    WRITE (6,360) (YS(J),J=1,IT)
    IF (IS9.EQ.1) GO TO 220
    WRITE (6,370) (ZTA(J),J=1,IT)
    IF (IPN.NE.0) WRITE (6,380)
    IF (FINAL) GO TO 230
    IT=C
    ITAB=ITAB+1
    GO TO 60
    CONTINUE
    ITH=ITHS
    CONTINUE
    IF (ITH.EQ.2.CR.IS9.EQ.ISW9B) GO TO 250
    ITH=2
    GO TO 10
    RETURN
250 C
    C
    C
260 FORMAT (1H0,43X,46H*****CONDITIONS ON WEDGE MODELS*****//WED 191
152X,A2,26HMODIFIED CHENG-KEMP THEORY) WED 192
270 FORMAT (45H MERGING EFFECTS SIGNIFICANT FOR XW LESS THAN.1PE10.2,8WED 193
    1H INCHES.) WED 194
280 FORMAT (65H STRONG-INTERACTION APPROXIMATION BREAKS DOWN FOR XW GRWED 195
    LEATER THAN.1PE10.2,8H INCHES.) WED 196
290 FORMAT (1H0) WED 197
300 FORMAT (31H0*****LEADING-EDGE RADIUS =.F6.3,25H INCH. ANGLE OWED 198
    IF ATTACK =.F5.1,26H DEGREES. CAPITAL GAMMA =.1PE10.2,9H. OMEGA =.WED 199
    2PE10.2,9H*****//WED 200
    FORMAT (12H PW (ATM),11X,1H=.1PE10.2,2) WED 201
    FORMAT (10H QWW (BTU/50 FT-SFC) =.1X,10(1PE10.2,A1)) WED 202
    FORMAT (22H QWW STANTON NUMMER,6X,1H=.1PE10.2,2) WED 203
    FORMAT (15H DELSTW (INCH),7X,1H=.1PE10.2,2) WED 204
    FORMAT (12H YS (INCHES),9X,1H=.1PE10.2,2) WED 205
    FORMAT (5H ZETA,16X,1H=.1PE10.2,2) WED 206
    FORMAT (5X,22H* FREE-MOLECULE LIMIT) WED 207
    END WED 208
    WED 209
    WED 210-

```

```

SUBROUTINE ...SOLN (ZETA,CAPGAM,ZZP,ZZPF,  

  SOLUTION OF THE CHENG EQUATION FOR FLOW OVER A WEDGE  

  IMPLICIT REAL*8 (A-H,O-Z)  

  KSEP= THIS IMPLICIT STATEMENT IN THE UNIVAC 1103 VERSION  

  LOGICAL CALLED  

  DATA CALLED /,FALSE/  

  DATA R0,31/1.81712C59D0,1.732C50800/  

  NAMELIST/WEFMDP/ XL,SOTL,Q1,Q2,Q3,Q4,Q5,ZT, DLDZT  

  IF (CALLED) GO TO 10  

  CALLED=.TRUE.  

  Q1=1.00  

  Q3=1.00/3.00  

  Q2=9.00/9.00  

  Q4=9.00  

  Q5=46.00/3.00  

  C103=10.00/3.00  

  C4=4.00  

  C32=1.500  

  C2=2.00  

  C769=75.00/9.00  

  P1=ZETA**C3  

  P2=DSORT(ZETA)  

  IF (ZETA.GT.1.D-8) GO TO 20  

  HODGE'S ANALYTICAL SOLUTION FOR SMALL ZETA  

  C12=Q1**2  

  Z1=.00096*P12+C.50869*P1*P2-0.02400*ZETA  

  XL=1.81711*P1+1.25975*P2+C.14712*P12  

  DLDZT=C.6057/P12+C.62988/P2+C.09808/P1  

  GO TO 30  

  ITERATIVE SOLUTION BASED ON CHENG/KEMP EXACT SOLUTION FOR GAMMA=0.  

  XLEB0=P1+R1*P2  

  XEB  

  IF (N.GT.20) GO TO 40  

  SOTL=DSORT(XL)  

  Q1=Q1+SOTL  

  Q2=Q1*Q1  

  Q3=Q1**2  

  Q4=Q2*Q2  

  Q5=QLOG(Q1)  

  ZT=CQ*Q4-C229*Q3+C9*Q2-C463*Q1+Q5*(C103-C4*SOTL+C2*Q5)+C769  

  ZEC2*(SOTL-XL/C2+C3*XL**C32-Q5)  

  DLDZT=C1/2  

  IF (CABS(Z1-ZETA)/ZETA.LE.1.D-4) GO TO 50  

  XL=XL-(Z1-ZETA)*DLDZT  

  XEB+1  

  GO TO 30  

  WRITE (6,60) ZETA,CAPGAM  

  WRITE (6,WEFMDP)  

  C950=CAPGAM**2  

  ZZP=XL+C950*ZETA  

  ZZPF=DLDZT+CGSO  

  ZEDSORT(Z**2+(CAPGAM*ZETA)**2)  

  RETURN  

  10  

  20  

  30  

  40  

  50  

  60

```

**ORIGINAL PAGE IS
OF POOR QUALITY**

-567-

WES 56
WES 57-

7HCAPGAM=E15.8)
END

ORIGINAL PAGE IS
OF POOR QUALITY

[illegible]

```

10      DO 30 K=1.50
      J=51-K
      DO 10 L=1.N
      IF (IS(L)-J) 10,20.10
      CONTINUE
      I(J)=JJ
      JJ=JJ-1
      GO TO 30
      I(J)=L
      CONTINUE
      FIND INDICES OF SPECIES TO BE USED IN LEWIS NUMBER CALCULATION.
      ID1=MINO(IATOM,IMOL)
      ID2=MAXO(IATOM,IMOL)
      CALL CXSECT (ID1,ID2)
      RE-INDEX CROSS SECTION DATA TO CORRESPOND TO ORDER OF INPUT
      SPECIES. SET Q(1,L,J)=1.0 FOR ALL PAIRS OF SPECIES FOR WHICH
      CROSS SECTION DATA ARE PROVIDED.
      DO 50 J=1.N
      DO 50 L=1.N
      Q(1,J,L)=0.
      Q(1,2,1)=1.
      NKQ=MINO(NNKQ,99)
      LQ=Q
      V(1)=1.
      KV=1
      KIJE=
      DO 90 K=1,NK2
      M=ISEQ(K)
      NK=KQ(M)
      KQ(K)=K
      NL=NV(NK)
      IF (NK.EQ.0) GO TO 70
      DO 60 KM=1,NK
      KV=KV+1
      V(KV)=V(KM,N)
      CONTINUE
      NK=NV(NK)
      LQ=LQ+NK
      NQ(K)=LO
      IF (NK.EQ.0) GO TO 90
      DO 80 KM=1,NK
      KIJE=KIJE+1
      L=L+Q(KM,N)
      J=J+Q(KM,N)
      CALL CXSECT (L,J)
      IF (J.LE.N.AND.L.LE.N) Q(1,L,J)=1.
      IQ(KIJE)=L
      JQ(KIJE)=J
      CONTINUE
      FOR KQ=9 OR 12. THE V-ARRAY DEPENDS ON THE INDEXING OF THE
      SPECIES. THESE DATA MUST ALSO BE RE-INDEXED.
      L=Q
      CALL BXSECT (L,MV,LL,J)
      IF (L.EQ.2) GO TO 110
      CALL CXSECT (LL,J)

```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

111 IF (J.LE.N.AND.LL.LE.N) O(I,LL,J)=1.
112 GO TO 100
113 EDIT CROSS SECTION DATA TO DELETE SPECIES NOT REQUIRED FOR
114 TRANSPORT CALCULATIONS.
115 LQ1=1
116 FOR KQ=0 OR 12, IT MAY BE NECESSARY TO COMPUTE CROSS SECTIONS FOR
117 SPECIES NOT INCLUDED IN TRANSPORT CALCULATIONS. STATEMENTS 120 TO
118 140 CHECK FOR THIS POSSIBILITY. IF SUCH CROSS SECTIONS ARE FOUND,
119 STATEMENTS 140 TO 300 REVISE THE DATA TO ELIMINATE THEM FROM THE
120 CALCULATIONS. STATEMENTS 300 THROUGH 380 THEN PROCEED TO EDIT THE
121 CROSS SECTION DATA TO REMOVE SPECIES NOT REQUIRED FOR THE
122 TRANSPORT CALCULATIONS.
123 CALL EXSECT (L,MV,LL,J)
124 IF (L.EQ.0) GO TO 300
125 IF (J.LE.N.AND.LL.LE.N) GO TO 120
126 IF (L.GT.1) LQ1=NQ(L-1)+1
127 LQ2=NQ(L)
128 IF (LQ1.GT.LQ2) GO TO 120
129 DO 130 LQ=LQ1,LQ2
130 IF (JQ(LQ).LF.N.AND.IQ(LQ).LE.N) GO TO 140
131 CONTINUE
132 GO TO 120
133 CROSS SECTIONS ARE REQUIRED FOR THE PAIR (LL,J). FIND UNUSED
134 STORAGE LOCATION FOR THEM.
135 DO 150 II=2,N
136 II1=II-1
137 DO 150 JJ=1,II1
138 IF (Q(I,II,JJ).EQ.0.) GO TO 190
139 CONTINUE
140 NO STORAGE SPACE AVAILABLE. ADD AN ADDITIONAL SPECIES TO CROSS
141 SECTION COMPUTATIONS.
142 IF (N.EQ.20) GO TO 540
143 N=N+1
144 JJ=MAX0(J,LL)
145 DO 160 J=1,50
146 IF (I(J).EQ.N) GO TO 170
147 CONTINUE
148 DO 180 K=1,50
149 IF (I(K).NE.JJ) GO TO 180
150 I(J)=JJ
151 I(K)=N
152 GO TO 40
153 CONTINUE
154 GO TO 540
155 STORE CROSS SECTIONS FOR PAIR (LL,J) IN LOCATION (II,JJ).
156 O(I,II,JJ)=1.
157 IT IS FIRST NECESSARY TO CONVERT ANY KQ=10 STEPS FOR THE PAIR TO
158 KQ=12.
159 NSV=NKQ
160 L=0
161 LQ1=0
162 L=L+1
163 IF (KQ(L).NE.10) GO TO 270
164 LQ2=NQ(L)
165 IF (L.GT.1) LQ1=NQ(L-1)

210	IF (LQ1,GE,LQ2) GO TO 270	XSE 166
	LQ=LQ1	XSE 167
	LV=LQ+1	XSE 168
	IV=IO(LQ)	XSE 169
	JV=JQ(LQ)	XSE 170
	IF ((LL,NE,J,AND,(IV,NE,LL,OR,JV,NE,J)).OR,(IV,NE,LL,AND,JV,NE,J))	XSE 171
	1 GO TO 260	XSE 172
C	CONVERSION TO KQ=12 MUST BE MADE FOR CURRENT STEP.	XSE 173
C	FIND LOCATION MM AT WHICH NEW DATA ARE TO BE INSERTED IN V-ARRAY.	XSE 174
	NKQ=L	XSE 175
	LV=0	XSE 176
220	CALL BXSECT (LV,MM,K,L)	XSE 177
C	IF (LV,NE,0) GO TO 220	XSE 178
	REVISE CROSS SECTION DATA	XSE 179
	MV=MMV	XSE 180
230	DO 230 K=MM,MMV	XSE 181
	V(MV+4)=V(MV)	XSE 182
	MV=MV-1	XSE 183
	V(MV+2)=IV	XSE 184
	V(MV+3)=V(MV+2)	XSE 185
	V(MV+4)=JV	XSE 186
	V(MV+5)=V(MV+4)	XSE 187
	MMV=MMV+4	XSE 188
	LQ1=LQ1+1	XSE 189
	LV=NSV	XSE 190
240	DO 240 K=NKQ,NSV	XSE 191
	KQ(LV+1)=KQ(LV)	XSE 192
	NQ(LV+1)=NQ(LV)	XSE 193
	LV=LV-1	XSE 194
	NSV=NSV+1	XSE 195
	NO(LV+1)=LQ1	XSE 196
	L=LV+2	XSE 197
	LV=LQ	XSE 198
	DO 250 K=LQ1,LQ	XSE 199
	LV=LV-1	XSE 200
250	IO(LV+1)=IO(LV)	XSE 201
	JQ(LV+1)=JQ(LV)	XSE 202
	IO(LQ1)=IV	XSE 203
	JQ(LQ1)=JV	XSE 204
	KQ(NKQ)=12	XSE 205
260	CONTINUE	XSE 206
270	IF (LQ,LT,LQ2) GO TO 210	XSE 207
	CONTINUE	XSE 208
	IF (L,LT,NSV) GO TO 200	XSE 209
	NKQ=NSV	XSE 210
C	CONVERSION OF KQ=10 TO KQ=12 COMPLETED. NOW REVISE DATA TO CHANGE	XSE 211
C	PAIR (LL,J) TO (II,JJ).	XSE 212
	LQ2=NQ(NKQ)	XSE 213
	DO 290 LQ=1,LQ2	XSE 214
	IF (IO(LQ),NE,LL,OR,JO(LQ),NE,J) GO TO 280	XSE 215
	IO(LQ)=II	XSE 216
	JQ(LQ)=JJ	XSE 217
280	CONTINUE	XSE 218
	L=0	XSE 219
290	CALL BXSECT (L,MV,IV,JV)	XSE 220

ORIGINAL PAGE IS
OF POOR QUALITY

```

C
C
300
310
320
330
340
350
360
370
380
390

IF (L.EQ.0) GO TO 110
IF (IV.NE.LL.OR.JV.NE.J) GO TO 290
JV=JJ
GO TO 290
EDIT CROSS SECTION DATA TO ELIMINATE SPECIES NOT REQUIRED IN
COMPUTATIONS.
DO 310 LQ=1,400
IF (IQ(LQ).GT.N.OR.JQ(LQ).GT.N) IQ(LQ)=0
CONTINUE
NEWL=0
NEWLO=0
NEWMV=1
MV1=2
NSV=NKQ
LQ1=1
KQ(100)=0
DO 370 L=1,NSV
K=KQ(L)
LQ2=NQ(L)
IF (LQ2.LT.LQ1) GO TO 330
DO 320 LQ=LQ1,LQ2
IF (IQ(LQ).NE.0) NEWLO=NEWLO+1
CONTINUE
LQ1=LQ2+1
IF (K.EQ.10) GO TO 350
IF (NEWLO.FQ.NQ(NEWL)) GO TO 360
MV2=MV1+NV(K)-1
DO 340 MV=MV1,MV2
NEWMV=NEWMV+1
V(NEWMV)=V(MV)
NEWL=NEWL+1
KQ(NEWL)=K
NQ(NEWL)=NEWLO
GO TO 370
NKQ=NKQ-1
MV1=MV1+NV(K)
LL=0
DO 380 LQ=1,400
IF (IQ(LQ).EQ.0) GO TO 380
LL=LL+1
IQ(LL)=IQ(LQ)
JQ(LL)=JQ(LQ)
CONTINUE
COMPUTE UNSPECIFIED NEUTRAL-NEUTRAL CROSS SECTIONS FROM EMPIRICAL
MIXING LAW. COMPUTATIONS ARE PERFORMED AT LAST SPECIFIED STEP
WITH KQ=10, OR, IF NO STEPS WITH KQ=10 ARE SPECIFIED, AFTER ALL
SPECIFIED CROSS SECTIONS HAVE BEEN COMPUTED.
LL=NKQ+1
DO 390 L=1,NKQ
IF (KQ(L).EQ.10) LL=L
CONTINUE
IF (LL.LE.NKQ) GO TO 400
KQ(LL)=10
NQ(LL)=NQ(LL-1)

```

```

400      NKQ=NKQ+1
      NN=0
      DO 420 L=1,ISS
      IF (IELEC.EQ.0) GO TO 410
      IZ(L)=LPIJ(L,1)
      IF (IZ(L).NE.0) GO TO 420
      NN=NN+1
      I(NN)=L
      IF (Q(I,L).EQ.0.) GO TO 540
      CONTINUE
      CALL AXSECT (LL,NN)
      FOR IONIZED SPECIES, USE EFFECTIVE COULOMB CROSS SECTIONS CHOSEN
      TO FIT TRANSPORT CALCULATIONS OF SPITZER.
      IF (LL.FQ.NKQ.AND.NQ(LL).EQ.NQ(LL-1)) GO TO 430
      NKQ=NKQ+1
      NQ(NKQ)=NQ(NKQ-1)
      DO 480 J=1,2
      IE=IELEC
      KK=J
      DO 470 JJ=1,2
      NN=0
      DO 450 L=1,ISS
      IF (IABS(IZ(L)))
      IF (L.FQ.IE.OR.(II.NE.J.AND.II.NE.KK)) GO TO 450
      NN=NN+1
      I(NN)=L
      CONTINUE
      CALL AXSECT (NKQ,NN)
      IF (NQ(NKQ).EQ.NQ(NKQ-1)) GO TO 460
      KQ(NKQ)=2
      NKQ=NKQ+1
      NQ(NKQ)=NQ(NKQ-1)
      V(NEWMV+1)=VCOUL(1,KK,J,JJ)
      V(NEWMV+2)=VCOUL(2,KK,J,JJ)
      NEWMV=NEWMV+2
      IF (KK.LE.1) GO TO 470
      KK=KK-1
      GO TO 440
      IE=0
      CONTINUE
      NN=0
      DO 490 L=1,ISS
      IF (IZ(L).EQ.0) GO TO 490
      NN=NN+1
      I(NN)=L
      CONTINUE
      CALL AXSECT (NKQ,NN)
      IF (NQ(NKQ).NE.NQ(NKQ-1)) GO TO 540
      SET UNSPECIFIED ION-NEUTRAL CROSS SECTIONS EQUAL TO TYPICAL VALUE
      ESTIMATED FOR N-O+ INTERACTION.
      DO 500 L=1,ISS
      I(L)=L
      CALL AXSECT (NKQ,ISS)
      IF (NQ(NKQ-1).NE.NQ(NKQ)) GO TO 510
      NKQ=NKQ-1

```

```

XSE 276
XSE 277
XSE 278
XSE 279
XSE 280
XSE 281
XSE 282
XSE 283
XSE 284
XSE 285
XSE 286
XSE 287
XSE 288
XSE 289
XSE 290
XSE 291
XSE 292
XSE 293
XSE 294
XSE 295
XSE 296
XSE 297
XSE 298
XSE 299
XSE 300
XSE 301
XSE 302
XSE 303
XSE 304
XSE 305
XSE 306
XSE 307
XSE 308
XSE 309
XSE 310
XSE 311
XSE 312
XSE 313
XSE 314
XSE 315
XSE 316
XSE 317
XSE 318
XSE 319
XSE 320
XSE 321
XSE 322
XSE 323
XSE 324
XSE 325
XSE 326
XSE 327
XSE 328
XSE 329
XSE 330

```

510 GO TO 520

K(NKQ)=6
V(NEWMV+1)=996.
V(NEWMV+2)=5.

520 CONTINUE

IF (NKQ.GT.100) GO TO 540

530 RETURN

INSUFFICIENT CROSS SECTION DATA AVAILABLE. TERMINATE CASE.

540 WRITE (6,550)

ERR=.TRUE.

WRITE (6,XSDMP)

GO TO 530

C
C

550 FORMAT (/116HOTTRANSPORT PROPERTIES OF DESIRED MIXTURE CANNOT BE CALCULATED FROM AVAILABLE DATA. REVISE CROSS SECTION INPUT DATA.)
END

331 XSE
332 XSE
333 XSE
334 XSE
335 XSE
336 XSE
337 XSE
338 XSE
339 XSE
340 XSE
341 XSE
342 XSE
343 XSE
344 XSE
345 XSE
346 XSE
347 XSE
348 XSE

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX

PROGRAMMER'S MANUAL FOR THE NOZFIT CODE

The present appendix documents the programming of the NOZFIT code, a small computer program for generating NATA-type curvefits to nozzle profiles. A user's manual for NOZFIT is presented in Appendix D of Volume II of this report (ref. 2).

Section A.1 of this appendix analyzes the NOZFIT main program and the subroutines used to produce plots of nozzle profile curvefits. Section A.2 is a glossary of Fortran symbols for NOZFIT. Finally, a source listing of NOZFIT and its associated subroutines is presented in Section A.3.

A.1 Analysis of NOZFIT Routines

This section discusses and explains the main program and each subroutine of the NOZFIT code. The purpose of this exposition is to provide an entry into the coding for programmers who wish to analyze errors, make corrections, or introduce modifications into the code.

A.1.1 NOZFIT Main Program

NOZFIT consists of two main sections. The first section computes the parameters of the profile fit from the input data (see Appendix D of Volume II); the second produces printed, punched, and plotted output.

(1) Computation of profile fit parameters. -- The parameters* of the curvefit to the nozzle or channel profile are calculated in the loop DØ 140, L = 1, 2. For L = 1, the parameters of the upstream sections are determined; for L = 2, the downstream sections are treated. For each region, the calculations start at the throat and proceed away from the throat. The index K enumerates the profile sections for

*The profile fit parameters and the analytical forms in which they appear are defined in Section 4.3 of Volume I (ref. 1).

each region in the order in which they are determined. The index J labels the sections for the entire nozzle, both upstream and downstream, in sequence starting at the upstream end. Finally, JP indexes the downstream boundaries of the profile sections in sequence starting at the upstream end.

The algorithm used to determine the parameters for each section depends upon the input "condition" index, ICØND(J).

a. ICØND(J) = 1

In each region (upstream or downstream), the first section treated is the one adjacent to the throat. This section is required to be a circular arc convex toward the axis (ISHAPE(J) = 2), and must be specified as a throat section (ICØND(J) = 1). For such a throat section, dy/dx must be zero at x = 0. Thus, equation I(122) shows that the abscissa P₂ of the circle center must be zero.

$$P_2 = 0 \quad (178)$$

The circle radius is equal to PAR(1,J):

$$P_3 = 2.54 \cdot \text{PAR}(1,J) \quad (179)$$

where the numerical factor converts from the inch unit used in the input to the centimeter unit used in the output. Finally, the ordinate of the circle center is equal to the throat radius plus the circle radius (see figure 28):

$$P_1 = P_3 + \frac{1}{2} \cdot 2.54 \cdot \text{DTH} \quad (180)$$

where DTH is the throat diameter in inches.

b. ICØND(J) = 2

For the condition ICØND(J) = 2, the current profile section is assumed to be straight (ISHAPE(J) = 1) and tangent to an adjacent circular section nearer the throat. The index of the section nearer the throat is denoted by INDEX. The profile parameters for this section are already known, because of the order which the sections are treated.

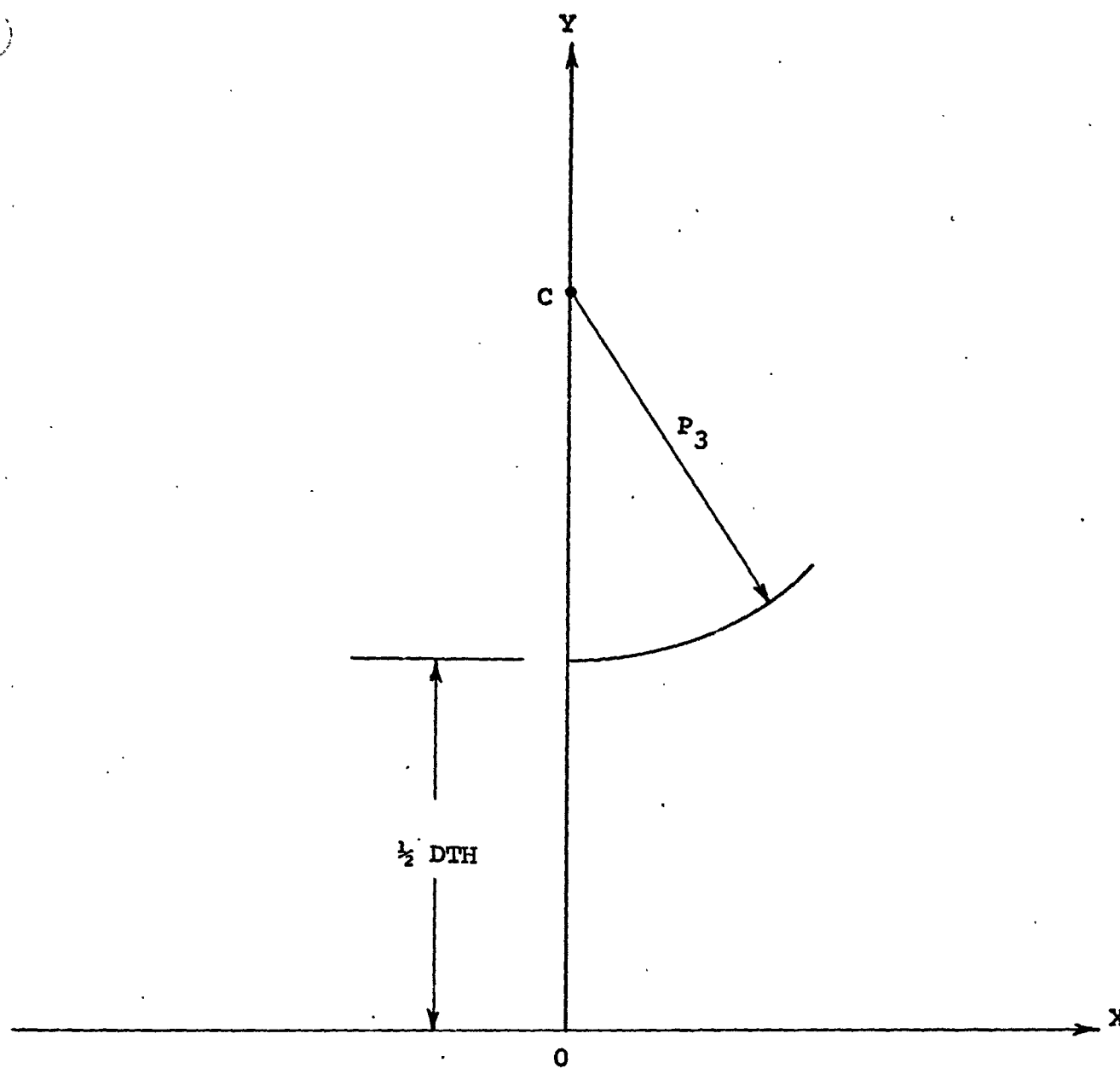


Figure 28. Geometry of the Condition $IC\emptyset ND(J) = 1$

Since the sections can be either upstream or downstream of the throat, and since the adjacent section INDEX can be either convex (ISHAPE(INDEX) = 2) or concave (ISHAPE(INDEX) = 3) toward the axis, there are four cases to be considered, as illustrated in figure 29. These cases are all treated in one set of formulas with the aid of the following symbols:

$$S_1 = \begin{cases} -1 & \text{upstream} \\ +1 & \text{downstream} \end{cases} \quad (181)$$

$$S_2 = \begin{cases} -1 & \text{for circle bottom, ISHAPE(INDEX) = 2} \\ +1 & \text{for circle top, ISHAPE(INDEX) = 3} \end{cases} \quad (182)$$

The symbol S_1 has the same sign as the slope of the current, straight section; S_2 is negative if the circle center C is above the line passing through the straight section, positive if the center is below.

The point of tangency of the straight and circular sections is marked by a "1" in each part of figure 29. The coordinates of this point are

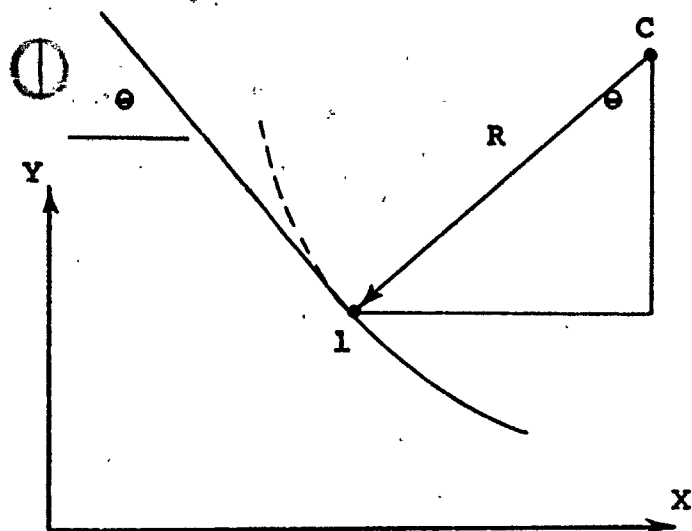
$$X_1 = X_C - S_1 S_2 R \sin \theta \quad (183a)$$

$$Y_1 = Y_C + S_2 R \cos \theta \quad (183b)$$

Here $R = \text{PARAM}(3, \text{INDEX})$ is the circle radius, and $\theta = \text{PAR}(1, J)$ is the angle of inclination of the straight section to the nozzle axis, taken positive. It is easy to show that the internal angle of the triangle at the vertex C is equal to θ . Also X_C and Y_C are the coordinates of the circle center in the adjacent circular section nearer the throat:

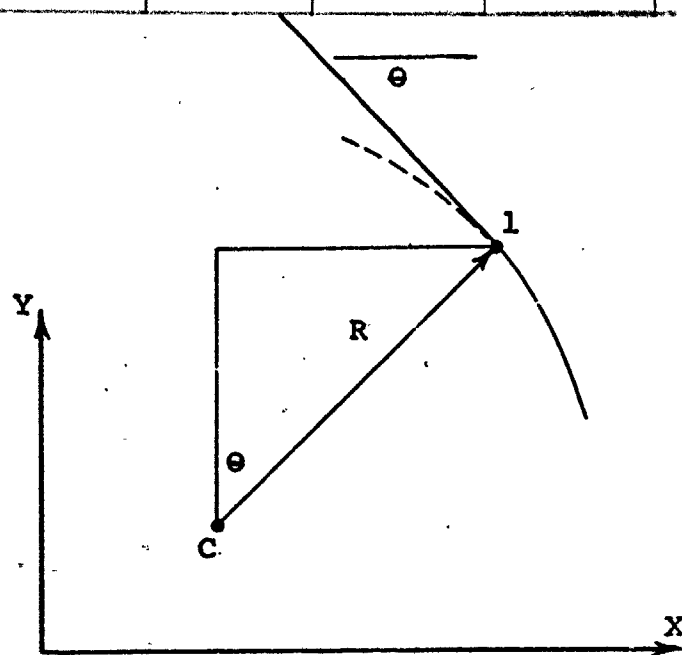
$$X_C = \text{PARAM}(2, \text{INDEX}) \quad (183c)$$

$$Y_C = \text{PARAM}(1, \text{INDEX}) \quad (183d)$$



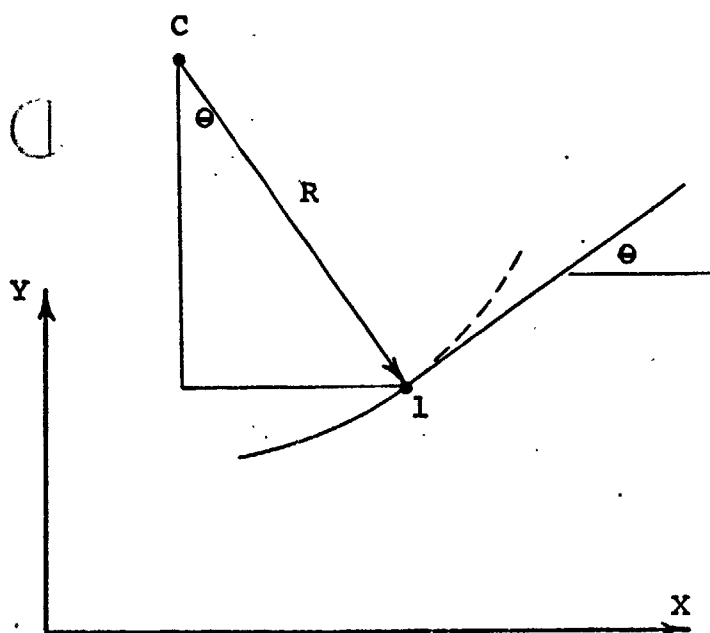
(a)

Upstream, ISHAPE(INDEX) = 2



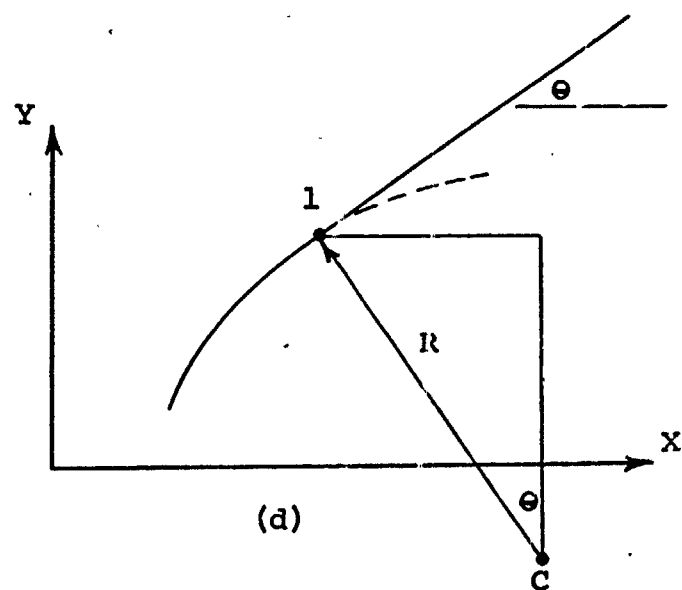
(b)

Upstream, ISHAPE(INDEX) = 3



(c)

Downstream, ISHAPE(INDEX) = 2



(d)

Downstream, ISHAPE(INDEX) = 3

Figure 29. Geometry of the Condition ICØND(J) = 2

The equation for the straight section is

$$Y - Y_1 = S_1 \tan \theta \cdot (X - X_1) \quad (184)$$

Thus, from equation I(121),

$$P_1 = Y_1 - S_1 X_1 \tan \theta \quad (185)$$

$$P_2 = S_1 \tan \theta \quad (186)$$

c. ICØND(J) = 3

For the condition ICØND(J) = 3, the current profile section is assumed to be circular and tangent to an adjoining straight section which is nearer the throat. The abscissa of the point of tangency is assumed to be PAR(2,J) (inches). As illustrated in figure 30, there are again four cases to be considered. As in the preceding case, the quantities S_1 and S_2 , equations (181) and (182), are used to treat all four cases in a single set of formulas.

In the present case, the coordinates of the point of tangency ("1" in figure 30) are known, since

$$X_1 = 2.54 \cdot \text{PAR}(2,J) \quad (187)$$

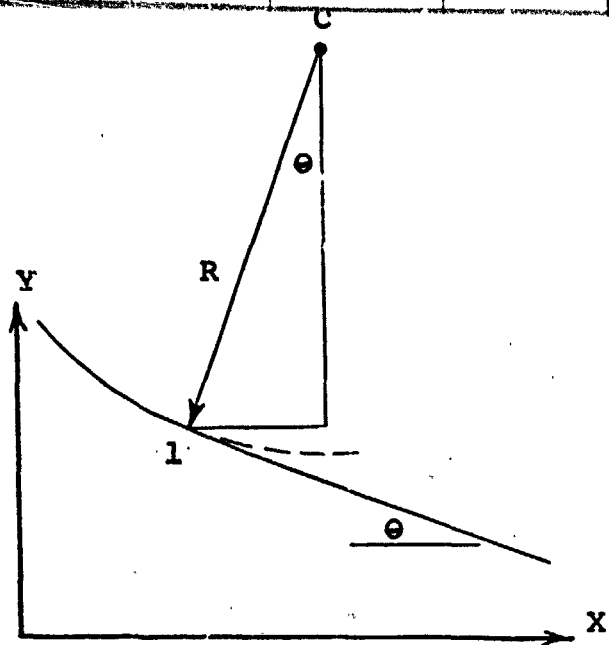
and Y_1 can be calculated from equation I(121) using the known parameter values for the adjoining straight section nearer the throat. The basic problem is to determine the coordinates (X_c , Y_c) of the circle center. Equations (183) are again applicable, and give

$$X_c = X_1 + S_1 S_2 R \sin \theta \quad (188a)$$

$$Y_c = Y_1 - S_2 R \cos \theta \quad (188b)$$

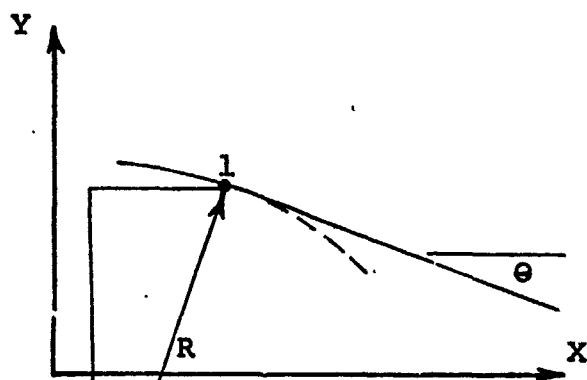
From equations I(122) and I(123), $P_2 = X_c$ and $P_1 = Y_c$. The radius P_3 is

①



(a)

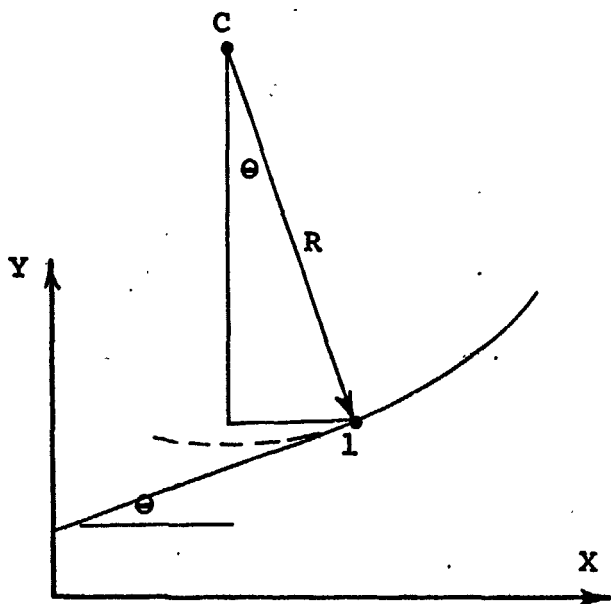
Upstream, ISHAPE(J) = 2



(b)

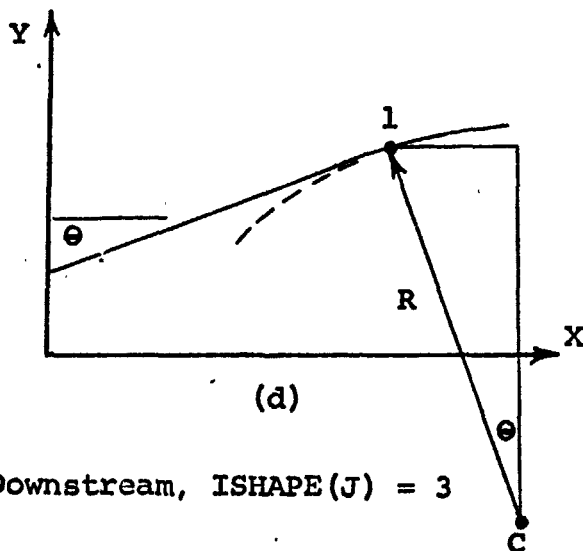
Upstream, ISHAPE(J) = 3

②



(c)

Downstream, ISHAPE(J) = 2



(d)

Downstream, ISHAPE(J) = 3

Figure 30. Geometry of the Condition ICØND(J) = 3

$$P_3 = 2.54 \cdot \text{PAR}(1,J) \quad (189)$$

$$d. \text{ ICØND}(J) = 4$$

The condition $\text{ICØND}(J) = 4$ allows the determination of a circular section which is tangent to two adjacent straight sections (J+1 and J-1). Figure 31 illustrates the four cases which must be considered. The straight section nearer the throat has the index INDEX; the one farther from the throat has the index INDX2. The angles $\theta_1 = \text{PAR}(1, \text{INDEX})$ and $\theta_2 = \text{PAR}(1, \text{INDX2})$ are known and the abscissa $X_1 = 2.54 \cdot \text{PAR}(2,J)$ of the intersection of the two lines is given. Since the curvefit parameters for the section INDEX have already been determined, the ordinate Y_1 of the intersection can be calculated from eq. I(121).

Since the circular section to be determined is tangent to both lines, its center C lies on the bisector of the angle between the two lines. The length of this bisector, from the intersection point "1" to the circle center "C", is denoted by L (or XL in the Fortran). Also, the angle between the bisector and either of the lines is denoted by ϕ . This angle is given by

$$\phi = \frac{1}{2} (\pi - |\theta_2 - \theta_1|) \quad (190)$$

since $(\pi - |\theta_2 - \theta_1|)$ is the included angle between the lines. Then, from either of the right triangles with a vertex at C,

$$L = R / \sin \phi \quad (191)$$

For $\text{ISHAPE}(J) = 2$ (parts (a) and (c) of figure 31), the angle CLA is equal to $\phi - \theta_1$. For $\text{ISHAPE}(J) = 3$ (parts (b) and (d) of figure 31), angle CLA is equal to $\phi - \theta_2$. Hence, for $\text{ISHAPE}(J) = 2$,

$$X_C = X_1 - S_1 L \cos (\phi - \theta_1) \quad (192a)$$

$$Y_C = Y_1 + L \sin (\phi - \theta_1) \quad (192b)$$

For $\text{ISHAPE}(J) = 3$,

$$X_C = X_1 + S_1 L \cos (\phi - \theta_2) \quad (193a)$$

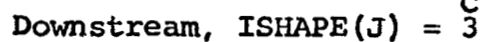
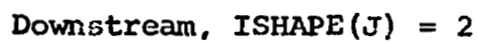


Figure 31. Geometry of the Condition. $1C3ND(5) = 4$

$$Y_c = Y_1 - L \sin (\phi - \theta_2) \quad (193b)$$

For either case, the distance from point "1" to the junction point "P" between the circular section and the straight section nearer the throat is equal to $L \cos \phi$. The projection of this distance onto the X-axis is $L \cos \phi \cos \theta_1$. Hence, the abscissa of the junction point is

$$X_p = X_1 - S_1 L \cos \phi \cos \theta_1 \quad (194)$$

It is clear from figure 31 that ISHAPE(J) can be equal to 2 only if $\theta_2 > \theta_1$ and ISHAPE(J) can be equal to 3 only if $\theta_1 > \theta_2$. If the user inputs a value for ISHAPE(J) violating these conditions, an error message is written and execution is terminated.

(2) Input and Output. -- NATA and its auxiliary programs are designed for operation on either a UNIVAC 1108 or an IBM System 360. So far as NØZFIT is concerned, the principal difference between the two computer systems is the word length. The UNIVAC 1108 contains 36-bit words which can store 6 alphanumeric characters. The IBM 360 contains 32- and 64-bit words which can store 4 or 8 alphanumeric characters, respectively. The plot routines used at NASA/JSC on the UNIVAC 1108 and at Avco Systems Division on the IBM 360/75 assume that arrays of alphanumeric information are close-packed (i.e., with no unused characters). To provide correct handling of the information on the "header" card*, NØZFIT contains read and write statements based on both the A6 format (for the UNIVAC) and the A4 format (for the IBM 360). The correct input/output statements for alphanumeric information are selected by reference to an indicator IWØRDL, which is set in subroutine FRAMEB (see below). Two versions of FRAMEB are available, one for use on the UNIVAC 1108 at the Johnson Space Center, the other for use on the IBM 360/75 at Avco. Apart from the differences in this one subroutine, NØZFIT

*See discussion of NØZFIT inputs in Appendix D of Volume II (ref. 2).

and its plot subroutines are fully compatible with both computer systems.

The coding of the printed and punched output of the profile curvefit parameters is straightforward. In the profile calculation (statements 240 to 310), the abscissa X is varied by increments of 0.03 inch from X = XSTART to X = XSTART + 6 inches. At each X, the index K of the profile section in which X lies is determined by finding the first K for which the downstream boundary ATP(K) is greater than X. The ordinate Y is then computed from equation I(121), I(122), or I(123) according as ISHAPE(K) is equal to 1, 2, or 3. The result of this calculation is in centimeters, and is converted into inches for output.

Finally, if the indicator PLOTS is .TRUE., the profile is plotted by a call to subroutine GRAPH. After the final case in the job (ENDJOB = .TRUE.), the entry FRAMB2 of subroutine FRAMEB is called. If the program is being run on the Avco IBM 360/75, FRAMB2 calls a library routine which closes the plot data set.

A.1.2 Subroutine FRAMEB

Subroutine FRAMEB contains all of the irreconcilable differences between the UNIVAC 1108 and IBM 360 versions of NØZFIT. Two versions of this routine are provided (Section A.3), one for each type of computer system.

The IBM 360 version sets the indicator IWØRDL to 2, to induce NØZFIT to use A4 formats in reading and writing alphanumeric information. It also calls the Avco library routine IDFRMV to produce an identification frame preceding the plots of nozzle profiles, if the control variable PLOTS is .TRUE.. When called through the entry FRAMB2, it calls the Avco library routine PLTND to empty the plot buffer and close the plotting data set.

The UNIVAC 1108 version of FRAMEB sets IWØRDL to 1, to induce NØZFIT to use A6 formats in reading and writing alphanumeric information. It performs no other function.

--	--	--	--	--	--	--

A plot identification frame is produced automatically on the 1108 computers at NASA/JSC when plots are requested on the JØB card, and the data set is also closed automatically.

A.1.3 Subroutine GRAPH

GRAPH is a general-purpose plotting routine which was developed for use in the auxiliary programs to the NATA code. It uses several lower level plot routines programmed by North American Aviation (NAA) as software for the S-C 4020 high-speed microfilm recorder. The capabilities of GRAPH may be summarized as follows:

- (1) It produces Cartesian graphs of Y versus X from data stored in arrays.
- (2) The data points can be displayed by up to 10 different plot symbols, or not displayed.
- (3) The data points can be connected by lines, or not connected. The lines can be continuous, dashed (with user-controlled dash length), dotted, or traced by a string of Charactron symbols.
- (4) Different data sets can be plotted on the same frame. The points from the different data sets can be represented by different plot symbols and/or connected by lines of different types.
- (5) A parameter value can be associated with each data set. The curves for different data sets can be labelled with numerical labels and a table displaying the labels and the associated parameter values can be shown in the margin to the right of the graph. Up to three distinct parameters, each with up to 10 values, can be included in a single plot.

The arguments in the calling sequence of GRAPH are defined in the Glossary of Fortran Symbols (Section A.2).

If a table of parameter values is to be printed in the frame, the first call to GRAPH must be preceded by a call to the NAA routine SETMIV (references 3, 17) to provide the required space in the right margin:

(ALL SETMIV (30, 125, 24, 24)

(1) Grid generation. -- The first part of GRAPH, down to statement 110, is executed only when the routine is called with $NF \neq 0$ to establish the grid for a new frame. In this case, the counter JCURVE for data sets plotted and the counter NPL for parameter tables are initialized to zero, and the NAA routine SMXYV (references 3, 17) is called to establish the logarithmic or linear mode for each axis as indicated by the GRAPH arguments LX, LY. Then the NAA routine DXDYV (references 3, 17) is called for each axis to determine the arguments DX, DY, NNM, MMM, III, JJJ, NXX, NYY of subroutine GRIDIV. If either axis is to be used as a logarithmic scale ($LX \neq 0$ or $LY \neq 0$), then the corresponding call to DXDYV is skipped, and DX and DY are set to 1 in accordance with instructions in references 3 and 17.

Beginning at statement 50, the parameters NXY(1) and NXY(2), which determine the numbers of characters to be displayed in the labels of vertical and horizontal grid lines, are established. The corresponding values NXX and NYY provided by DXDYV were found to be unsatisfactory in some applications. For each axis, if \log_{10} of the maximum absolute value to be plotted is greater than 5 or less than -3, the grid-line labels are printed in scientific notation, with three significant digits, by setting $NXY(I) = -3$. Otherwise, a fixed-point format is used, with a number of characters determined from the magnitude of the maximum value to be plotted.

Next, the GRIDIV arguments III and JJJ are reversed in sign to force placement of the grid line labels outside

of the grid area. Also, if numerical labels for the individual curves are to be displayed, the lower limits XL and YL are decreased slightly, and the upper limit YU is increased slightly, to allow space for the labels within the grid area. Then GRIDLV is called to draw the grid and to establish scaling factors for subsequent use in plotting. Finally, the counter ICURVE for curves in the first family (i.e., for various values of the first parameter, if any) is initialized to zero.

(2) Plotting of a data set. -- The second part of GRAPH plots an individual data set, that is, the data contained in the arrays X and Y for a single call to GRAPH. This section of the subroutine extends from statement 110 down to statement 190.

First, the counters ICURVE and JCURVE are each incremented by 1. If ICURVE is equal to 1 after this increase, the counter NPL for parameter tables is also increased by 1. Note that ICURVE is equal to 1 at this point if this is the first call to GRAPH for a new frame, or if ICURVE was set to zero in the calling routine before the current call to GRAPH. Setting ICURVE to 0 in the calling routine is a signal to GRAPH that a new family of curves with a new parameter is being started.

Next, the NAA routines NXV and NYV are used to initialize IX1 and IY1 to the raster coordinates corresponding to X(1) and Y(1), respectively. Then the plot symbol to be used in displaying the data points is selected on the basis of the arguments NC(1) and NC(2). If NC(1) is negative, the symbol index NSYMBL is set equal to ICURVE, so that a new symbol from the standard table in PØINTB is used for the points on each successive curve in a family of curves. If NC(1) is positive, NSYMBL is set equal to NC(1); this option allows direct user control of the plot symbol. For NC(1) = 0, the individual points are not plotted.

The DØ loop from statement 140 to statement 180 plots the data points and draws the connecting lines, if any. First, subroutine PØINTB (Section A.1.4 of this appendix) is called to plot the current (Ith) data point. Then I2 is set equal to I+1. If LINE(1) is nonzero, subroutine LINEB (Section A.1.5 of this appendix) is called to draw the line segment from the Ith to the I2th point. The type of line to be drawn (continuous, dashed, etc.) is determined by the argument LINE in the call to LINEB. Then IX1 and IX2 are set equal to the raster coordinates corresponding to X(I2) and Y(I2), respectively, using the NAA function routines NXV and NYV, for use in calling PØINTB the next time around the loop.

When these plotting operations have been completed, if JCURVE is equal to 1, the NAA routine CHSIZV is called with arguments (3,3) to restore the normal character size.* Then subroutine RITE2V (references 3, 17) is called to write the abscissa and ordinate labels ABSL and ØRDL and the plot title TITL in the bottom, left, and top margins of the plot, respectively.

(3) Parameter tables and lables. -- If the argument LL is zero, then parameter tables and curve labelling are not to be used, and a RETURN is executed at statement 190 in GRAPH. If LL is 1, the parameter tables are produced but the curve labelling is skipped. If LL is neither 0 nor 1, then the curves are labelled also. To implement the labelling, the code calls the NAA routines NXV and NYV to determine the raster coordinates, NXV1 and NYV1, of the first point on the curve. It then computes the raster coordinates, IX1 and IY1, of a point which lies 16 raster units away from the first point of the curve in a direction opposite to that in which the second point lies. It then writes the curve label (CURVE = ICURVE) into the plot by calling the NAA subroutine LABLV (references 3, 17).

*A nonstandard character size is set in PØINTB by calling CHSIZV(3,2), to obtain plot symbols whose height and width are approximately equal.

The tables of parameter values are produced by the coding from statement 200 to statement 300. For ICURVE = 1, the code first writes the table heading. The heading consists of a sample of the type of line used for the current family of curves, and a two-line printed specification of the nature of the current parameter. The sample line is produced by calling the entry LINEBR of subroutine LINEB (discussed below). It is horizontal and lies at the raster ordinate IY2. The printed heading is produced by calling the NAA routine PRINTV. Its first line lies 15 raster units below the sample line, and its second line is an additional 20 raster units lower. The tables for different parameters (families of curves) have a vertical separation of 300 raster units.

The body of the parameter table consists of rows each containing the numerical label for the curve (equal to CURVE), a sample of the plot symbol used, and the parameter value. The numerical label is omitted for LL = 1. When it is included, it is produced by calling LABLV. The sample plot symbol is produced by calling PØINTB, and the parameter value by another call to LABLV. The arguments NCHAR and NDMAX of LABLV are determined as functions of IPVM, which is $\log_{10} (PVAL)$ rounded down in magnitude to the nearest integer. If IPVM is greater than 4 or less than -2, NCHAR is set to -3 to give scientific notation with three significant figures. Otherwise, fixed-point notation is used, and NCHAR and NDMAX are adjusted to provide the required number of characters and positions to the left of the decimal point.

A.1.4 Subroutine LINEB

Subroutine LINEB draws a straight line between two specified points (X1, Y1), (X2, Y2). When it is called through its entry LINEBR, the two points are specified in raster coordinates (MX1, MY1), (MX2, MY2). The argument LLINE is a three-element array whose values control the type of line produced and its darkness, as defined in the Glossary of Fortran Symbols (Section A.2).

① The raster coordinates of the two points to be connected by a line are denoted, within LINEB, by (NX1, NY1), (NX2, NY2). If entry was through LINEBR, these values are set equal to MX1, etc. If entry was through LINEB, the raster coordinates are determined by calling the NAA subroutines NXV and NYV.

For $LINE \equiv LLINE(1) = 1$, the points are connected with a continuous line by calling NAA subroutine LINEV. This call is executed IDARK times to obtain the desired darkness.

① For $LINE > 1$, beginning in statement 50, the sine (SINA) and cosine (COSA) of the angle between the vector from point 1 to point 2 and the positive X-axis are computed. Then, for $LINE = 2$, a dashed line is drawn from point 1 to point 2, with the dash length in millimeters* being approximately equal to $IPAR \equiv LLINE(3)$. The distance DEL from the beginning of one dash to the beginning of the next is calculated, in raster units, as $10.8 \times IPAR$. The numerical coefficient in this formula is obtained by noting that the frame width is 1024 raster units and is 190 mm in hard copy. Thus, 1 mm on the hard copy is equivalent to 5.4 raster units, the dash length HDEL is equal to $5.4 \times IPAR$, and DEL is twice as large, allowing for the gap between two successive dashes. The number of dashes to be used in the line joining the two points is thus $NP = DIST/DEL$, where DIST is the separation of the points in raster units. If the two points are too close together in comparison with the specified dash length, integer arithmetic gives $NP = 0$. This possibility should be borne in mind by the user of GRAPH.

LINEB next computes the increments IDX and IDY in raster coordinates between the beginning and end of a single dash, using the previously determined trigonometric

*This description of the dash length is applicable to hard copy from the Calcomp 890 CRT Plotter at Avco, or to enlargements of microfilm plots produced on the SD 4060 at NASA/JSC to a frame size of 7.5 x 7.5 inches.

functions COSA and SINA. It then calls the NAA subroutine LINE2V (references 3, 17) NP times to draw the dashed line joining the end points specified in the arguments. In the DØ loop, DØ 70 I = 1, NP, the quantities VX1 and VY1 represent the desired starting points for each dash in raster coordinates. These quantities are calculated using floating point arithmetic to avoid the loss of accuracy possible in integer arithmetic, and are rounded down to obtain the integer values NX1, NY1 used in the calls to LINE2V.

For LINE = 3, the line joining the specified end points is produced by plotting a string of Charactron symbols. The symbol to be used is specified by IPAR, which is used as the index in the POINTV character table (references 3, 17). The following list is excerpted from references 3 and 17.

<u>IPAR</u>	<u>Character</u>
0	.
1	O
2	x
4	Y
5	+
6	*
10	H
15	⊗
22	Ø
25	I

The separation of successive points, DEL, is set to 5 raster units if IPAR = 0 (i.e., if the plotting character is a dot), and is set to 15 raster units otherwise. The number of characters to be used in tracing the line is then $NP = DIST/DEL$. The string of characters is then produced by calling the four-argument version of PØINTV (references 3, 17) NP times, in the loop DØ120 I = 1, NP. The minus sign on the argument IPAR of PØINTV suppresses the plotting of a central dot in each character.

A.1.5 Subroutine POINTB

This routine plots a point at the raster coordinates (IX, IY), using a plot symbol selected from a table ICHAR(I) of ten standard symbols. The index NSYMBL in the calling sequence specifies which symbol is to be used. The symbols are as follows:

<u>I</u>	<u>Symbol</u>
1	X
2	+
3	O
4	*
5	H
6	C
7	∅
8	Y
9	Λ
10	I

These symbols are produced by calling the NAA subroutine RITE2V (references 3, 17). The inversions for I = 7 and 9, and the 90-degree rotation for I = 10, are accomplished using the argument K(NSYMBL) of RITE2V. Before RITE2V is called, a call is made to CHSIZV (references 3, 17) to make the height of the symbols nearly equal to their width instead of 50 percent greater, as is normally the case. The argument IDARK of POINTB and RITE2V controls the number of times the symbol is overdrawn and thus its darkness in the plot.

A.2 Glossary of Fortran Symbols for NOZFIT

This section defines the Fortran symbols used in the NOZFIT main program and in its associated subroutines. The variables are listed in alphanumeric order within each routine, and the routines themselves are in alphabetic order. There are no common variables.

A.2.1 Main Program

ABSL(I)	Array containing alphanumeric abscissa label for plots
ACØN	Conversion factor ($180/\pi$) from radians to degrees
ATP(JP)	Axial coordinate at the downstream boundary of the JPth profile section (cm)
CARDS	Logical control; .FALSE. value suppresses punched card output
CØMMNT(I)	Hollerith descriptive data
CTHETA	Cos (THETA)
DTH	Throat diameter (inch)
DX	Increment in X for calculation of profile (inch)
ENDJØB	Logical control; .TRUE. value terminates the job at the end of the current case
FISHAP(I)	Floating point value representing ISHAPE(I) in punched-card output
FNSECT(I)	NSECTS(I) + 0.1 for card output
I	DØ index
IC	ICØND(J) for the Jth profile section
ICØND(J)	Index specifying the type of condition to be used in determining the Jth profile section (see input discussion, Appendix D of Volume II, ref. 2 or Section A.1.1 of the present Appendix)
ICURVE	Unused argument of subroutine GRAPH
INCR(L)	0 for L = 1; 1 for L = 2

INDEX	Index of the profile section adjacent to the current (Jth) section and nearer the throat
INDX2	Index of the profile section adjacent to the current (Jth) section and farther from the throat
ISGN(L)	1 for L = 1; -1 for L = 2
ISH	ISHAPE(INDEX)
ISHAPE(J)	Indicator for shape of Jth profile section (see input discussion, Appendix D of Volume II)
ISJ	ISHAPE(K)
IWØRDL	Indicator for number of characters per word on the type of computer system being used: <div style="margin-left: 100px;"> IWØRDL = 1 6-character word (UNIVAC 1108) IWØRDL = 2 4-character word (IBM 360) </div>
J	Profile section index
JP	Index of profile section boundaries
J1	2K - 1; first J value on the Kth card containing PARAM(I,J) data in the punched output
J2	J1 + 1; second J value on the Kth card containing PARAM(I,J) data in the punched output
K	DØ index; profile section index in calculation of profile from curvefit
KP	Continuation card counter punched in column 6 of output cards

L	Indicator for upstream profile sections (L = 1) and downstream sections (L = 2)
LINE(I)	Argument of subroutine GRAPH (see below)
LL	Argument of subroutine GRAPH (see below)
NC(I)	Argument of subroutine GRAPH (see below)
NØZZLE	Profile index for use in NATA
NS	NSECTS(L) in computation of profile fit parameters; total number of sections in coding of output
NSM1	NS-1
NSECTS(L)	Number of profile sections upstream (L = 1) and downstream (L = 2) of the throat
ØRDL(I)	Alphanumeric label for ordinate in plot
PAR(I,J)	Input parameter values for the Jth profile section; see discussion of inputs, Appendix D of Volume II
PARAM(I,J)	Parameters of curvefit for Jth profile section; see discussion of outputs, Appendix D of Volume II
PHI	$0.5 (\pi - \text{THETA2} - \text{THETA1})$; see discussion of ICØND = 4 in Section A.1.1.
PI	3.14159265
PLABL(I)	Argument of subroutine GRAPH (see below)
PLØTS	Logical control, .FALSE. value suppresses plot output
PVAL	Argument of subroutine GRAPH (see below)

R	Circle radius for a circular-arc section (cm)
RTHCM	Throat radius in centimeters
STHETA	Sin (THETA)
S1	-1.0 for L = 1, +1.0 for L = 2
S2	0.0 for ISHAPE(J) = 1; -1.0 for ISHAPE(J) = 2; +1.0 for ISHAPE(J) = 3
S2ARR(ISH)	Array containing S2 values for the three shapes
THETA	Angle of inclination of a straight section to the nozzle axis (radians)
THETA1	For ICØND = 4, inclination angle of the adjacent section closer to the throat
THETA2	For ICØND = 4, inclination angle of the adjacent section farther from the throat
TITLE(I)	Title for plots (Hollerith)
TTHETA	Tan(THETA)
X	Axial coordinate (cm)
XC	X-coordinate of the circle center for the adjacent section nearer the throat
XINCH(I)	Axial coordinate (inches)
XL	$R/\sin(\text{PHI})$; see discussion of ICØND = 4 in Section A.1.1
XLØ	Lower limit of abscissa in plot
XSTART	Left limit on X for calculating nozzle profile (inch)
XUP	Upper limit of abscissa in plot

XZERØ	Axial coordinate of start of boundary layer, for use in NATA (cm)
XZERØI	XZERØ expressed in inches
XI	Intermediate X values in calculations of parameters for ICØND = 2, 3, and 4
Y	Profile ordinate (cm)
YC	Y-coordinate of the circle center for the adjacent section nearer the throat
YINCH(I)	Profile ordinate (inches)
YLØ	Lower limit on ordinate in plot
YUP	Upper limit on ordinate in plot
YI	Intermediate Y values in calculations of parameters for ICØND = 2, 3, and 4

A.2.2 Subroutine FRAMEB

BIN	Alphanumeric bin number for plot identification frame (Avco)
IWØRDL	Defined above (Main program)
PLØTS	Defined above (Main program)
XMEMØ	Alphanumeric memo number for plot identification (Avco)
XNAME	Alphanumeric submitter name for plot identification frame (Avco)

A.2.3 Subroutine GRAPH

ABSL	Array containing 48-character label for abscissa scale
CURVE	ICURVE expressed as a floating-point value
DC	Parameter controlling the spacing of grid lines on linear scales; DC = 16 is a normal value. The spacing increases or decreases with the value of DC
DX	Floating point data increment at which vertical grid lines are displayed
DY	Floating point data increment at which horizontal grid lines are displayed
DYL	Amount by which YL is decreased and YU increased to allow space for numerical labels of curves
HYP	Distance (in raster units) between the first two points of a curve
I	00 index
ICURVE	Counter for the number of curves plotted on the same frame. If this argument is set to zero by the calling program, a new parameter label for a new series of curves is printed in the margin; this feature is used only when two or more families of curves are to be plotted in the same frame.
IDXV	Interval in raster units between the abscissas of the first two points in a curve
IDYV	Interval in raster units between the ordinates of the first two points of a curve

IERR	Error indicator in subroutine DXDYV (references 3, 17); if a nonzero value of IERR is returned, grid generation is impossible with the given upper and lower limits on X and Y
III	Argument of subroutine GRIDLV (references 3, 17). Every IIIth vertical line of the grid is labelled with a numerical value; a negative III forces placing the labels outside the grid
IMAG(I)	For I = 1 and 2, \log_{10} of the largest numerical value to be plotted on the abscissa and ordinate, respectively
IPVM	\log_{10} of the absolute magnitude of a parameter value (PVAL)
IXI	Abscissa of the location where a curve label is to be printed (raster units)
IY	Ordinate of the location where a parameter value is to be printed in the margin (raster units)
IY1	Ordinate of the location where a curve label is to be printed (raster units)
IY2	Ordinate of the location where a sample of the type of line used for a family of curves is drawn in the margin (raster units)
I2	$I + 1$
J	D ϕ index
JCURVE	Counter for the total number of curves plotted in a given frame

JJJ	Argument of subroutine GRIDLV (references 3, 17). Every JJJth horizontal line of the grid is labelled with a numerical value; a negative JJJ forces placing the labels outside the grid
LINE(I)	Control parameters for lines joining data points, defined as follows:
LINE(1)	0 Do not join points with lines 1 Join points with continuous lines 2 Join points with dashed lines 3 Join points with strings of characters
LINE(2)	Number of strikes for line; use 2 for dark lines
LINE(3)	Ignored if LINE(1) is 0 or 1 Length of dashes in mm if LINE(1) = 2 Index of character in POINTV table if LINE(1) = 3 (see Procedures Manual, Computation and Analysis Division, MSC, p. 5.c.7.5.1.127) (Use LINE(3) = 0 with LINE(1) = 3 to produce dotted lines)
LL	Control parameter for labelling of curves: LL = 0 Suppress labelling of curves and printing of parameter values in table at right of grid LL = 1 Print parameter values and plot symbols but do not label curves LL = 2 Label curves and print parameter values and plot symbols
LX	Argument of subroutine SMXYV (references 3, 17); 0 sets linear mode for abscissa, nonzero log mode

LY	Argument of subroutine SMXYV (references 3, 17); 0 sets linear mode for ordinate, nonzero log mode
MMM	Argument of subroutine GRIDLV (references 3, 17). Every MMMth horizontal grid line is retraced for emphasis
NC(I)	For I = 1 and 2, control parameters for plotting points: NC(1) < 0 Use standard table of 10 characters for plotting points, a new character for each successive call to GRAPH for the same family of curves in the same frame (see subroutine PØINTB) NC(1) = 0 Supress plotting of points NC(1) > 0 Use the NC(1)th character in the standard table (see PØINTB) NC(2) Number of strikes for each symbol plotted; use 1 for light characters, 2 for dark
NCHAR	Argument of subroutine LABLV (see references 3, 17); number of characters to be displayed, including leading blanks and the decimal point, if any
NC1	NC(1)
NC2	NC(2)
NDMAX	Argument of subroutine LABLV (see references 3, 17); maximum number of characters to be displayed to the left of the decimal point

NF	Control parameter for frame advance: NF = 0 Plot data on same frame NF = 1 Start a new frame, establish a grid, and plot the data on the new frame
NLAST	Argument of subroutine RITE2V (references 3, 17); not used in GRAPH
NNN	Argument of subroutine GRID1V (references 3, 17). Every NNNth vertical grid line is retraced for emphasis
NP	Number of data points to be plotted
NPL	Counter for families of curves plotted in the same frame
NSYMBL	Index of point symbol in standard table (see PØINTB)
NXV1	Abscissa of first point in raster coordinates
NXX	Argument of subroutine DXDYV (references 3, 17); not used in GRAPH
NXY(I)	For I = 1 and 2, the number of characters to be displayed in the labels of vertical and horizontal grid lines; arguments of subroutine GRID1V (references 3, 17)
NYV1	Ordinate of first point in raster coordinates
NYY	Argument of subroutine DXDYV (references 3, 17); not used in GRAPH
ØRDL(I)	Alphanumeric label for ordinate
PLABL(I,J)	For J = 1, 2, a two-line alphanumeric label for the parameter

PVAL	Parameter value
TITL(I)	Alphanumeric title
X(I)	Array containing the abscissas of the data points
XL	Lower limit on X for all data to be plotted in a given frame
XU	Upper limit on X for all data to be plotted in a given frame
XYMAG(I)	For I = 1 or 2, the largest absolute magnitude abscissa or ordinate value to be plotted in a given frame
Y(I)	Array containing the ordinates of the data points
YL	Lower limit on Y for all data to be plotted in a given frame
YU	Upper limit on Y for all data to be plotted in a given frame

A.2.4 Subroutine LINEB

CØSA	Cosine of the angle between the vector to be drawn and the positive X-axis
DEL	Twice the dash length in raster units
DIST	Distance between the beginning and end of the vector in raster units
HDEL	Dash length in raster units
I	DØ index over the dashed or plot symbols used to draw the vector
IDARK	Number of times lines or symbols are drawn

IDX	Increment in the abscissa for a dash (raster units)
IDY	Increment in the ordinate for a dash (raster units)
IPAR	Length of dashes in millimeters for LINE = 2; integer (NS) for selecting the plot symbol from the table in subroutine PØINTV (references 3, 17) for LINE = 3
J	DØ index for double-strike plotting to obtain dark lines or symbols
LINE	Control parameter for type of line: LINE = 1 Continuous line LINE = 2 Dashed line LINE = 3 String of Charactron symbols
LLINE(I)	LLINE(1) = LINE LLINE(2) = IDARK LLINE(3) = IPAR
MX1	Abscissa of the start point in raster units
MX2	Abscissa of the end point in raster units
MY1	Ordinate of the start point in raster units
MY2	Ordinate of the end point in raster units
NP	Number of dashes in vector
NX1	Abscissa of the start point in raster units
NX2	Abscissa of the end point in raster units
NY1	Ordinate of the start point in raster units
NY2	Ordinate of the end point in raster units

RASTER	Logical indicator, .TRUE. for entry through LINEBR, .FALSE. for entry through LINEB
SINA	Sine of the angle between the vector to be drawn and the positive X-axis
SQ	Square of the distance between the start and end points of the vector (raster units)
VX1	Floating point value of NX1
VY1	Floating point value of NY1
X1	Abscissa of the start point
X2	Abscissa of the end point
Y1	Ordinate of the start point
Y2	Ordinate of the end point

A.2.5 Subroutine PØINTB

CHAR(I)	Array of characters for plotting points
IDARK	Argument of RITE2V controlling darkness of plotted symbol (see references 3, 17)
IX	Abscissa of location where point is plotted (raster coordinate)
IY	Ordinate of location where point is plotted (raster coordinate)
K(I)	Argument of RITE2V controlling orientation of plotted symbol; used to present each of the symbols "H", "C", and "Y" in two orientations
NLAST	Argument of RITE2V (references 3, 17); not used in PØINTB

NSYMBL

Index used to select plotting symbol from
CHAR array

A.3 Listing of NØZFIT

Presented below are source program listings for the main program of NØZFIT, and for subroutines FRAMEB, GRAPH, LINEB, and PØINTB. Listings are provided for both the IBM (Avcc) and UNIVAC (NASA/JSC) versions of FRAMEB.

ORIGINAL PAGE IS
OF POOR QUALITY

4540N PROGRAM FOR DETERMINING AREA R=0 PARAMETERS FOR NATA

LOGICAL ENDJOB,CARDS,PLOTS
DIMENSION ISHAPE(12),PARAM(3,12),ATP(11),PLABL(6)
DIMENSION FNSECT(2),FISHPAR(12)
DATA PI /3.14159265/, CARDS,PLOTS /2*,TRUE./
DIMENSION PAR(2,12),ICOND(12),NSECTS(2)
DIMENSION INCR(2), S2ARR(3)
NAMLIST /INPUT/, NSECTS,ISHAPE,PAR,ICOND,DTH,ENDJOB,XSTART,
1 XZEROI,NOZZLE,CARDS,PLOTS

DEFINITIONS OF INPUTS

NSECTS(1) NUMBER OF UPSTREAM SECTIONS IN NOZZLE PROFILE
NSECTS(2) NUMBER OF DOWNSTREAM SECTIONS IN NOZZLE PROFILE
NOTE-- NSECTS(1) + NSECTS(2) MUST BE .LE. 10
ISHAPE(J) INDEX SPECIFYING SHAPE OF J-TH SECTION (COUNTING FROM
UPSTREAM INLET)

ISHAPE(J)=1 STRAIGHT SECTION
ISHAPE(J)=2 CIRCLE BOTTOM
ISHAPE(J)=3 CIRCLE TOP

PARAMETER VALUES FOR J-TH SECTION
FOR ISHAPE(J)=1, PAR(1,J)=ANGLE OF INCLINATION TO
NOZZLE AXIS IN DEGREES (POSITIVE VALUE)
FOR ISHAPE(J)=2 OR 3, PAR(1,J)=CIRCLE RADIUS IN INCHES
THROAT DIAMETER IN INCHES

DTH=180./PI
ACOND(J)

INDEX SPECIFYING CONDITION DEFINING J-TH SECTION
ICOND(J)=1 THROAT CONDITION (Y=1.27*DTH AND DY/DX=0
AT X=0)

ICOND(J)=2 STRAIGHT SECTION (J) IS TANGENT TO ADJACENT
CIRCULAR SECTION NEARER THE THROAT
ICOND(J)=3 CIRCULAR SECTION (J) IS TANGENT TO ADJACENT
STRAIGHT SECTION (NEARER THE THROAT)

ICOND(J)=4 CIRCULAR SECTION (J) IS USED TO BREAK A
SHARP ANGLE BETWEEN TWO ADJOINING STRAIGHT
SECTIONS WHICH INTERSECT AT X=2.54*PAR(2,J)
LEFT LIMIT ON X FOR CALCULATING NOZZLE PROFILE (INCH)
INLET POSITION IN INCHES ABOVE THE THROAT (NEGATIVE)
NOZZLE INDEX FOR NATA
LOGICAL CONTROL - FALSE VALUE SUPPRESSES PUNCH OUTPUT
LOGICAL CONTROL - FALSE VALUE SUPPRESSES PLOT
LOGICAL CONTROL FOR JOB TERMINATION

DIMENSION ORDL(12),ABSL(12),NC(2),LINE(3)
DIMENSION COMMENT(18),XINCH(201),YINCH(201),TITLE(12)
DATA NSECTS,ISHAPE,ICOND /26*C/, PAR /24*0./
DATA ABSL /4* . . . X . . . INCH,6* . . . /
DATA ORDL /4* . . . Y . . . INCH,6* . . . /

EQUIVALENCE(COMMENT(1),TITLE(1))
DATA ENDJOB /0,TRUE./, INCR /0,1/
DATA S2ARR /0,1,1./
CALL FRAHER (IWCORL,PLOTS)
IF (IWCORL.FO.1) GO TO 20
READ (5,330) (COMMENT(I),I=1,18)

GO TO 30
 READ (5,340) (CCVMT(I),I=1,12)
 FIRST 4 CHARACTERS ON THIS CARD SHOULD BE NAME OF FACILITY FOR USE NOZ
 IDENTIFICATION IN NATA (DCA, EOS, ETC.) NOZ
 READ (5,INPUT) NOZ
 WRITE (6,INPUT) NOZ
 ZERO ARRAYS NOZ
 DO 40 I=1,12 NOZ
 IF (I.NE.12) ATP(I)=0. NOZ
 DO 40 J=1,3 NOZ
 PARAM(J,I)=0. NOZ
 COMPUTE PROFILE FIT PARAMETERS
 DO 140 L=1,2 NOZ
 L=1 UPSTREAM NOZ
 L=2 DOWNSTREAM NOZ
 NS=NSCTS(L) NOZ
 DO 130 K=1,NS NOZ
 IF (L.EQ.2) GO TO 50 NOZ
 SI=-1. NOZ
 J=NS-K+1 NOZ
 ISGN=1 NOZ
 GO TO 60 NOZ
 J=NSCTS(1)+K NOZ
 ISGN=-1 NOZ
 SI=1. NOZ
 IC=ICGND(J) NOZ
 JP=J-INCR(L) NOZ
 INDEX=J+ISGN NOZ
 GO TO (70,80,90,100), IC NOZ
 IF (ISHAPE(J).NE.2) GO TO 120 NOZ
 THRCAT SECTION (ICOND=1) NOZ
 PARAM(3,J)=2.54*PAR(1,J) NOZ
 PARAM(2,J)=0. NOZ
 PARAM(1,J)=PARAM(3,J)+1.27*DTN NOZ
 ATP(JP)=C. NOZ
 GO TO 130 NOZ
 IF (ISHAPE(J).NE.1) GO TO 120 NOZ
 IF (ISHAPE(INDEX).NE.2.AND.ISHAPE(INDEX).NE.3) GO TO 120 NOZ
 CONDITION ICOND=2 NOZ
 R=PARAM(3,INDEX) NOZ
 THETA=PAR(1,J)/ACON NOZ
 ISH=ISHAPE(INDEX) NOZ
 S2=S2ARR(ISH) NOZ
 XC=PARAM(2,INDEX) NOZ
 YC=PARAM(1,INDEX) NOZ
 STHETA=SIN(THETA) NOZ
 CTHETA=COS(THETA) NOZ
 TTHETA=STHETA/CTHETA NOZ
 X1=XC-S1*S2*RR*STHETA NOZ
 Y1=S2*R*CTHETA+YC NOZ
 PARAM(1,J)=Y1-S1*X1*TTTHETA NOZ
 PARAM(2,J)=S1*TTTHETA NOZ
 PARAM(3,J)=0. NOZ

ORIGINAL PAGE IS
 OF POOR QUALITY

```

ATP(JP)=X1
GO TO 130
IF (ISHAPE(J).NE.2.AND.ISHAPE(J).NE.3) GO TO 120
IF (ISHAPE(INCX).NE.1) GO TO 120
CONDITION ICOND=3
R=2.54*PAR(1,J)
X1=2.54*PAR(2,J)
IC=1SHAPE(J)
S2=S2ARP(ISH)
THETA=PAR(1,INDEX)/ACON
Y1=PARAM(1,INDEX)+PARAM(2,INDEX)*X1
PARAM(1,J)=Y1-S2*R*COS(THETA)
PARAM(2,J)=X1+S1*S2*R*SIN(THETA)
PARAM(3,J)=R
ATP(JP)=X1
GO TO 130
IF (ISHAPE(J).NE.2.AND.ISHAPE(J).NE.3) GO TO 120
IF (ISHAPE(INDEX).NE.1) GO TO 120
INDX2=J-1SGN
IF (ISHAPE(INDX2).NE.1) GO TO 120
CONDITION ICOND=4
R=2.54*PAR(1,J)
X1=2.54*PAR(2,J)
THETA1=PAR(1,INDEX)/ACON
THETA2=PAR(1,INDEX2)/ACON
PHI=C*(PI-ABS(THETA2-THETA1))
IF (THETA2.GT.THETA1.AND.ISHAPE(J).EQ.3) GO TO 120
IF (THETA2.LT.THETA1.AND.ISHAPE(J).EQ.2) GO TO 120
Y1=PARAM(1,INDEX)+PARAM(2,INDEX)*X1
XL=R/SIN(PHI)
ATP(JP)=X1-S1*XL*COS(PHI)*COS(THETA1)
PARAM(3,J)=R
IF (ISHAPE(J).EQ.3) GO TO 110
PARAM(2,J)=X1-S1*XL*COS(PHI-THETA1)
PARAM(1,J)=Y1+XL*SIN(PHI-THETA1)
GO TO 130
PARAM(2,J)=X1+S1*XL*COS(PHI-THETA2)
PARAM(1,J)=Y1-XL*SIN(PHI-THETA2)
GO TO 130
WRITE (6,350) J
CALL EXIT
CONTINUE
CONTINUE

OUTPUT
PRINTED OUTPUT
IF (IAORDL.EQ.1) GO TO 150
WRITE (6,360) (COMMT(I),I=1,18)
GO TO 160
WRITE (6,370) (COMMNT(I),I=1,12)
WRITE (6,380)
WRITE (6,390)
NS=NSECTS(1)+NSECTS(2)
DO 180 J=1,NS

```

[illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

```

170 IF (J.EQ.NS) GO TO 170
180 WRITE (6,400) J,ATP(J), (PARAM(I,J), I=1,3)
182 GO TO 180
C CONTINUE
IF (.NOT. CARDS) GO TO 240
PUNCH=1,27*DTM
RTHCM=2,54*XZERC
XZERC=2,54*XZERC
DO 190 I=1,2
  FNSECT(I)=FNSECTS(I)+0.1
190 DO 200 I=1,12
  IF (I.GT.NS) ISHAPE(I)=0
  FISHP(I)=ISHAPE(I)+0.1
  WRITE (6,420)
  IF (I.EQ.DL.EQ.1) GO TO 210
  PUNCH 520, (COMMENT(I), I=1,16), NOZZLE
  WRITE (6,440) (COMMENT(I), I=1,16), NOZZLE
  GO TO 220
  PUNCH 510, (COMMENT(I), I=1,11), NOZZLE
  WRITE (6,430) (COMMENT(I), I=1,11), NOZZLE
  PUNCH 530, NOZZLE, RTHCM, XZERC, (FNSECT(I), I=1,2), NOZZLE
  WRITE (6,450) NOZZLE, RTHCM, XZERC, (FNSECT(I), I=1,2), NOZZLE
  PUNCH 540, (FISHP(I), I=1,12), NOZZLE
  WRITE (6,460) (FISHP(I), I=1,12), NOZZLE
  PUNCH 550, (ATP(I), I=1,6), NOZZLE
  WRITE (6,470) (ATP(I), I=1,6), NOZZLE
  PUNCH 560, (ATP(I), I=7,11), NOZZLE
  WRITE (6,480) (ATP(I), I=7,11), NOZZLE
  DO 230 K=1,6
    J1=2*K-1
    J2=J1+1
    KP=X+3
    WRITE (6,490) KP, (PARAM(I,J), I=1,3), J=J1, J2, NOZZLE
    PUNCH 570, KP, (PARAM(I,J), I=1,3), J=J1, J2, NOZZLE
    PUNCH 580, COMMENT(1), NOZZLE
    WRITE (6,500) COMMENT(1), NOZZLE
    PRINT OUT OF PROFILE
    WRITE (6,590)
    WRITE (6,600)
    DX=C.C
    DO 310 I=1,201
      XINCH(I)=XSTART+(I-1)*DX
      X=2.54*XINCH(I)
      NSM=NS-1
      DO 250 J=1,NSM
        K=J
        IF (ATP(J).GT.X) GO TO 260
        CONTINUE
        X=NS
        ISJ=ISHAPE(K)
        GO TO (270,280,290), ISJ
        Y=PARAM(1,K)+PARAM(2,K)*X
        GO TO 300
        Y=PARAM(1,K)-SORT(PARAM(3,K)*#2-(X-PARAM(2,K))*#2)
250
260
270
280

```

```

290 GO TO 300
300 Y=PARAM(1,K)+SQRT(PAPAM(3,K)**2-(X-PARA(2,K))**2)
WRITE (6,610) XINCH(1),YINCH(1)
310 CONTINUE
IF (.NOT.PLOTS) GO TO 320
PLOT OF PROFILE
NC(1)=0
NC(2)=1
LINE(1)=1
LINE(2)=2
LL=0
XLO=XINCH(1)
XUP=XINCH(201)
YLO=0.
YUP=0.
CALL GRAPH (1,0,0,201,XINCH,YINCH,"LO,XUP,YLO,YUP,16,NC,LINE,ORDL
320 1,AUSL,TITLE,LL,VAL,PLABL,ICURVE)
IF (.NOT.ENDJOB) GO TO 10
IF (PLOTS) CALL FRAMB2
CALL EXIT
330 FORMAT (18A4)
340 FORMAT (12A6)
350 FORMAT (30H) INCOMPATIBLE DATA FOR SECTION,13)
360 FORMAT (1H1,18A4,/)
370 FORMAT (1H1,12A6,/)
380 FORMAT (20H NOZZLE PROFILE PARAMETERS)
390 FORMAT (1H,8X,1H,17X,3HATP,10X,10HPARAM(1,J),10X,10HPARAM(2,J),1
400 1X,10HPARAM(3,J))
410 FORMAT (110,1P4E20,6)
420 FORMAT (110,20X,1P3E20,6)
430 FORMAT (110,20H DATA CARDS PRODUCED/)
440 FORMAT (1X,1HC,5X,11A6,1X,4HN0ZZ,13)
450 FORMAT (1X,1HC,5X,16A4,3X,4HN0ZZ,13)
460 FORMAT (7X,7HDATA ZP,12,2H /,F10,5,1H,,F10,5,1H,,F6,1,1H,,F6,1,1H,
1,20X,4HN0ZZ,13)
470 FORMAT (6X,1H1,6X,12(F4,1,1H,),1X,4HN0ZZ,13)
480 FORMAT (5X,1H2,6(F10,6,1H,),1X,4HN0ZZ,13)
490 FORMAT (6X,1H3,5(F10,6,1H,),12X,4HN0ZZ,13)
500 FORMAT (6X,1H4,6X,2H4H,A4,1H/,54X,4HN0ZZ,13)
510 FORMAT (6X,1H5,11A6,1X,4HN0ZZ,13)
520 FORMAT (1HC,5X,16A4,3X,4HN0ZZ,13)
530 FORMAT (6X,7HDATA ZP,12,2H /,F10,5,1H,,F10,5,1H,,F6,1,1H,,F6,1,1H,
1,20X,4HN0ZZ,13)
540 FORMAT (5X,1H1,6X,12(F4,1,1H,),1X,4HN0ZZ,13)
550 FORMAT (5X,1H2,6(F10,6,1H,),1X,4HN0ZZ,13)
560 FORMAT (5X,1H3,5(F10,6,1H,),12X,4HN0ZZ,13)
570 FORMAT (5X,1H4,6X,2H4H,A4,1H/,54X,4HN0ZZ,13)
580 FORMAT (5X,1H5,11A6,1X,4HN0ZZ,13)
590 FORMAT (1HC,5X,16A4,3X,4HN0ZZ,13)
600 FORMAT (1HC,12A,7HXY(INCH),13X,7HY(INCH))
610 FORMAT (1P2E20,3)
END
NOZ 221
NOZ 222
NOZ 223
NOZ 224
NOZ 225
NOZ 226
NOZ 227
NOZ 228
NOZ 229
NOZ 230
NOZ 231
NOZ 232
NOZ 233
NOZ 234
NOZ 235
NOZ 236
NOZ 237
NOZ 238
NOZ 239
NOZ 240
NOZ 241
NOZ 242
NOZ 243
NOZ 244
NOZ 245
NOZ 246
NOZ 247
NOZ 248
NOZ 249
NOZ 250
NOZ 251
NOZ 252
NOZ 253
NOZ 254
NOZ 255
NOZ 256
NOZ 257
NOZ 258
NOZ 259
NOZ 260
NOZ 261
NOZ 262
NOZ 263
NOZ 264
NOZ 265
NOZ 266
NOZ 267
NOZ 268
NOZ 269
NOZ 270
NOZ 271
NOZ 272
NOZ 273
NOZ 274
NOZ 275-

```

SUBROUTINE FRAMED (IWORDL,PLOTS)
LOGICAL PLOTS

FRAMES FOR AVCO IBM 360/75
PRUDUCES IDENTIFICATION FRAME FOR PLOTS AND CLOSES PLOT DATA SET

```

DIMENSION XNAME(3),XMEMO(2)
DATA XNAME /' ','BADE',, '%, XMEMO /'NOZF',,IT',/. BIN /'29',/
IWORDL=2
IF (PLOTS) CALL IDFRMV (XNAME,BIN,XMEMO)
RETURN
ENTRY FRAMB2
CALL PLTND
RETURN
END

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15-
FRI FRI FRI FRI FRI FRI FRI FRI FRI FRI FRI FRI FRI FRI FRI

ORIGINAL PAGE IS
OF POOR QUALITY

CC
CC
CC

ROUTINE FRAMES (IWORDL.PLOTS)

FRAMES FOR UNIVAC 1108

DUMMY ROUTINE

ENTRY FRAMB2

IWORDL=1

RETURN

END

1 2 3 4 5 6 7 8 9
FRU FRU FRU FRU FRU FRU FRU FRU FRU

ORIGINAL PAGE IS
OF POOR QUALITY

CCCC

```

100 SUBROUTINE GRAPH (NF,LX,LY,NP,X,Y,XL, ,YL,YU,DC,NC,LINE,ORDL,ABSL,GRA
110 1,TITL,LL,PVAL,PLABL,ICURVE)
120 DIMENSION X(120),Y(120),LINE(3),CROL(12),ABSL(12),TITL(12),
130 1 PLABL(3,2),NXY(2),IMAG(2),XYMAG(2),NC(2)
140 NOTE - 48 CHARACTERS IN ABSCISSA, ORDINATE, AND TITLE LABELS
150 NOTE - 12 CHARACTERS IN EACH OF 2 LINES OF PARAMETER LABEL (PLABL)
160 NOTE - CALL SETMIV (30,125,24,24) BEFORE FIRST CALL TO GRAPH,
170 IF PARAMETER TABLE IS TO BE PRINTED
180 IF (NF.EQ.0) GO TO 110
190 JCURVE=0
200 NPL=0
210 CALL SMXYV (LX,LY)
220 IF (LX.EQ.0) GO TO 10
230 DX=1.
240 NNN=1
250 GO TO 20
260 CALL DXDYV (1,XL,XU,DX,NNN,111,NXX,DC,IERR)
270 IF (LY.EQ.0) GO TO 30
280 DY=1.
290 NMM=1
300 GO TO 40
310 CALL DXDYV (2,YL,YU,DY,MMM,JJJ,NYY,DC,IERR)
320 IF (LX.NE.0.AND.LY.NE.0) GO TO 50
330 IF (IERR.EQ.0) GO TO 50
340 WRITE (6,310) XL,XU,YL,YU
350 RETURN
360 XYMAG(1)=AMAX1(ABS(XL),ABS(XU))
370 XYMAG(2)=AMAX1(ABS(YL),ABS(YU))
380 DO 90 I=1,2
390 IMAG(I)=ALOG10(XYMAG(I))
400 IF (IMAG(1).GT.5.OR.IMAG(1).LT.-3) GO TO 80
410 IF (IMAG(1)) 60,60,70
420 NXY(I)=I*ABS(IMAG(I))+4
430 GO TO 90
440 NXY(I)=IMAG(I)+1
450 GO TO 90
460 NXY(I)=3
470 CONTINUE
480 III=-III
490 JJJ=-JJJ
500 IF (LL.LT.2) GO TO 100
510 XL=XL-0.022*(XU-XL)
520 YL=Y-0.022*(YU-YL)
530 YL=YL-DYL
540 YU=YU+DYL
550 CALL GRIDIV (1,XL,XU,YL,YU,DX,DY,NNN,MMM,111,JJJ,NXY(1),NXY(2))
560 JCURVE=0
570 ICURVE=ICURVE+1
580 IF (ICURVE.EQ.1) NPL=NPL+1
590 JCURVE=JCURVE+1
600 IXI=NXY(X(1))
610 IYI=NXY(Y(1))
620 NC=NC(2)
630 NC1=NC(1)
640 IF (NC1) 120,140,130

```



```

120 NSYMBL=ICURVE
130 GO TO 140
140 NSYMBL=NC1
150 DO 180 I=1,NP
160 IF (NC1.EQ.0) GO TO 150
170 CALL POINTB (IX1,IY1,NSYMBL,NC2)
180 IF (I.EQ.NP) GO TO 180
190 I2=I+1
200 IF (LINE(I)) 170,170,160
210 CALL LINEB (X(I),Y(I),X(I2),Y(I2),LINE)
220 IX1=NXV(X(I2))
230 IY1=NYV(Y(I2))
240 CONTINUE
250 IF (JCURVE.GT.1) GO TO 190
260 CALL CHSIZV (3,3)
270 CALL RITE2V (80,9,1023,90,2,48,1,ABSL,NLAST)
280 CALL RITE2V (9,80,1023,180,2,48,1,ORDL,NLAST)
290 CALL PITE2V (80,1012,1023,90,2,48,1,TITL,NLAST)
300 IF (LL.EQ.0) RETURN
310 CURVE=ICURVE
320 IF (LL.EQ.1) GO TO 200
330 NXV1=NXV(X(I))
340 NYV1=NYV(Y(I))
350 IDXV=NXV(X(I))-NXV1
360 IDYV=NYV(Y(I))-NYV1
370 HYP=IDXV**2+IDYV**2
380 HYP=SQRT(HYP)
390 IX1=NXV1-16.*IDXV/HYP
400 IY1=NYV1-16.*IDYV/HYP
410 CALL LABLV (CURVE,IX1,IY1,2,2,2)
420 IF (ICURVE.GT.1) GO TO 230
430 IY=950-(NPL-1)*300
440 IY2=IY+15
450 CALL LINEBR (930,IY2,1023,IY2,LINE)
460 DO 220 I=1,2
470 DO 210 J=1,2
480 CALL PRINTV (I2,PLABL(I,I),924,IY)
490 IY=IY-20
500 IY=IY-20
510 IF (LL.EQ.1) GO TO 240
520 CALL LABLV (CURVE,900,IY,2,1,2)
530 CALL POINTB (924,IY,NSYMBL,NC2)
540 IF (PVAL.EQ.0) GO TO 250
550 IPVM=ALOG10(ABS(PVAL))
560 GO TO 260
570 IPVM=0
580 IF (IPVM.GT.4.OR.IPVM.LT.-2) GO TO 290
590 IF (IPVM) 270,270,280
600 NCHAR=IABS(IPVM)+5
610 NDMAX=1
620 GO TO 300
630 NCHAR=5
640 NDMAX=IPVM+1
650 GO TO 300
660 NDMAX=1

```

GRA 56
 GRA 57
 GRA 58
 GRA 59
 GRA 60
 GRA 61
 GRA 62
 GRA 63
 GRA 64
 GRA 65
 GRA 66
 GRA 67
 GRA 68
 GRA 69
 GRA 70
 GRA 71
 GRA 72
 GRA 73
 GRA 74
 GRA 75
 GRA 76
 GRA 77
 GRA 78
 GRA 79
 GRA 80
 GRA 81
 GRA 82
 GRA 83
 GRA 84
 GRA 85
 GRA 86
 GRA 87
 GRA 88
 GRA 89
 GRA 90
 GRA 91
 GRA 92
 GRA 93
 GRA 94
 GRA 95
 GRA 96
 GRA 97
 GRA 98
 GRA 99
 GRA 100
 GRA 101
 GRA 102
 GRA 103
 GRA 104
 GRA 105
 GRA 106
 GRA 107
 GRA 108
 GRA 109
 GRA 110

NCHAR=-3
CALL LABLV (PVAL,940.IY,NCHAR,1,NDMAX)
RETURN

300

C

310

FORMAT (51HOGRID GENERATION IMPOSSIBLE WITH THE PARAMETERS XL=,1PEGRA 116
110.3.SH, XU=.E10.3.5H, YL=.E10.3.5H, YU=.E10.3)
END

GRA 111
GRA 112
GRA 113
GRA 114
GRA 115
GRA 116
GRA 117
GRA 118-

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE POINTB (IX,IY,NSYMBL,IDARK)
DIMENSION CHAR(10),K(10)
DATA CHAR /'X','+','D','*','H','C','Y','Y','Y','H' /
DATA K /6*90,270,90,270,0 /
CALL CHSIZV (3,2)
CALL RITE2V (IX,IY,1023,K(NSYMBL),IDARK,1,1,CHAR(NSYMBL),NLAST)
RETURN
END

```

1 2 3 4 5 6 7 8-
 POI
 POI
 POI
 POI
 POI
 POI
 POI

ORIGINAL PAGE IS
 OF POOR QUALITY

```

C      SUBROUTINE LINEB (X1,Y1,X2,Y2,LLINE)
C      LLINE=LLINE(1), IDARK=LLINE(2), IPAR=LLINE(3)
C      X1 = ABSCISSA OF FIRST POINT
C      Y1 = ORDINATE OF FIRST POINT
C      X2 = ABSCISSA OF SECOND POINT
C      Y2 = ORDINATE OF SECOND POINT
C      LLINE = 1 CONNECT POINTS WITH A CONTINUOUS LINE
C      LLINE = 2 CONNECT POINTS WITH A DASHED LINE
C      LLINE = 3 CONNECT POINTS WITH A STRING OF CHARACTER SYMBOLS
C      IF LLINE=1 IPAR IS IGNORED
C      IF LLINE=2 IPAR = LENGTH OF DASHES IN MILLIMETERS
C      IF LLINE=3 IPAR = INTEGER (NS) SELECTING PLOT SYMBOL FROM
C      POINTV TABLE
C      IDARK = NUMBER OF TIMES LINE OR SYMBOLS ARE DRAWN
C      LOGICAL RASTER
C      DIMENSION LLINE(3)
C      RASTER=.FALSE.
C      GO TO 1C
C      ENTRY LINEBR(MX1,MY1,MX2,MY2,LLINE)
C      RASTER=.TRUE.
C      LLINE=LLINE(1)
C      IF (LLINE.LE.0) RETURN
C      IDARK=LLINE(2)
C      IPAR=LLINE(3)
C      IF (.NOT. RASTER) GO TO 20
C      NX1=MX1
C      NY1=MY1
C      NX2=MX2
C      NY2=MY2
C      GO TO 3C
C      NX1=NXV(X1)
C      NX2=NXV(X2)
C      NY1=NYV(Y1)
C      NY2=NYV(Y2)
C      IF (LLINE.GT.1) GO TO 50
C      DO 40 I=1,IDARK
C      CALL LINEV (NX1,NY1,NX2,NY2)
C      RETURN
C      SQ=(NX2-NX1)**2+(NY2-NY1)**2
C      DIST=SQRT(SQ)
C      VX1=NX1
C      VY1=NY1
C      COSA=(NX2-NX1)/DIST
C      SINA=(NY2-NY1)/DIST
C      IF (LLINE.GT.2) GO TO 8C
C      DEL=10.6*IPAR
C      NP=DIST/DEL
C      HDL=0.5*DEL
C      IDX=HDL*COSA
C      IDY=HDL*SINA
C      DO 70 I=1,NP
C      DO 60 J=1,IDARK
C      CALL LINE2V (NX1,NY1,IDX,IDY)
C      IF (I.EQ.NP) GO TO 70
C      VX1=VX1+DEL*COSA

```

```

LIN 1
LIN 2
LIN 3
LIN 4
LIN 5
LIN 6
LIN 7
LIN 8
LIN 9
LIN 10
LIN 11
LIN 12
LIN 13
LIN 14
LIN 15
LIN 16
LIN 17
LIN 18
LIN 19
LIN 20
LIN 21
LIN 22
LIN 23
LIN 24
LIN 25
LIN 26
LIN 27
LIN 28
LIN 29
LIN 30
LIN 31
LIN 32
LIN 33
LIN 34
LIN 35
LIN 36
LIN 37
LIN 38
LIN 39
LIN 40
LIN 41
LIN 42
LIN 43
LIN 44
LIN 45
LIN 46
LIN 47
LIN 48
LIN 49
LIN 50
LIN 51
LIN 52
LIN 53
LIN 54
LIN 55

```

56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76-

LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN
LIN

VY1=VY1+DEL*SINA
NX1=VX1
NY1=VY1
CONTINUE
RETURN
IF (IPAR.EQ.0) GO TO 90
DEL=15.
GO TO 100
DEL=5.
NP=DIS/DEL
DO 120 I=1,NP
DO 110 J=1,IDARK
CALL POINTV (NX1,NY1,-IPAR,1.)
IF (I.EQ.NP) GO TO 120
VX1=VX1+DEL*COXA
VY1=VY1+DEL*SINA
NX1=VX1
NY1=VY1
CONTINUE
RETURN
END

70
80
90
100
110
120

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES

1. Bade, W. L.; and Yos, J. M.: The NATA Code - Theory and Analysis, Volume I. NASA CR-2547, 1975.
2. Bade, W. L.; and Yos, J. M.: The NATA Code - User's Manual, Volume II. NASA CR-141743, 1975.
3. Anon.: Programmers' Reference Manual, S-C 4020 High-Speed Microfilm Recorder. Doc. No. 9500056, Stromberg-Carlson Division of General Dynamics, Oct. 1964.
4. UNIVAC Data Processing Division: 1108 Multi-Processor System, EXEC II Programmers Reference Manual. Univac Division of Sperry Rand Corporation UP-4058, 1966, pp. 3-20 to 3-33.
5. Dewey, C. F., Jr.; and Gross, J. F.: Exact Similar Solutions of the Laminar Boundary Layer Equations. Rep. RM-5089-ARPA, RAND Corp., 1967.
6. Lordi, J. A.; Mates, R. E.; and Moselle, J. R.: Computer Program for the Numerical Solution of Nonequilibrium Expansions of Reacting Gas Mixtures. NASA CR-472, May 1966.
7. International Business Machines Corporation: System/360 Scientific Subroutine Package, Version III, Programmer's Manual, Program No. 360A-CM-03X, Fifth Ed., August 1970.
8. Carnahan, B.; Luther, H. A.; and Wilkes, J. O.: Applied Numerical Methods. John Wiley & Sons, 1969, pp. 282-291.
9. Devoto, R. S.: Transport Coefficients of Partially Ionized Argon. Phys. Fluids, vol. 10, no. 2, Feb. 1957, pp. 354-364.

10. Mason, E. A.; Vanderslice, J. T.; and Yos, J. M.: Transport Properties of High-Temperature Multi-component Gas Mixtures. Phys. Fluids, vol. 2, no. 6, Nov.-Dec. 1959, pp. 688-694.
11. Monchick, L.: Collision Integrals for the Exponential Repulsive Potential. Phys. Fluids, vol. 2, no. 6, Nov.-Dec. 1959, pp. 695-700.
12. Monchick, L.; and Mason, E. A.: Transport Properties of Polar Gases. J. Chem. Phys., vol. 35, no. 5, Nov. 1961, pp. 1676-1697.
13. Kihara, T.; Taylor, M. H.; and Hirschfelder, J. O.: Transport Properties for Gases Assuming Inverse Power Intermolecular Potentials. Phys. Fluids, vol. 3, no. 5, Sept.-Oct. 1960, pp. 715-720.
14. Winovich, W.: On the Equilibrium Sonic-Flow Method for Evaluating Electric-Arc Air Heater Performance. NASA TN D-2132, Mar. 1964.
15. Leipman, H. W.; and Roshko, A.: Elements of Gas Dynamics. John Wiley & Sons, 1957.
16. Boger, R. C.; and Aiello, G. F.: Hypersonic Flow on Yawed Wedges with Leading-Edge Bluntness and Viscous Interaction. J. Spacecraft and Rockets, vol. 8, no. 7, July 1971, pp. 729-735.
17. Anon: Procedures Manual, Sec. 5.c.7.5.0-SD 4060 System. Computation and Analysis Division - MSC, NASA Manned Spacecraft Center, Apr. 1969.